



TUGAS AKHIR - KI141502

Rancang Bangun Sistem Cerdas Berbasis Cloud Pada Studi Kasus Sistem Monitoring Daya *Charging Station* Mobil Listrik

MUHAMMAD FARID WAJDI
NRP 5112100194

Dosen Pembimbing I
Dr.Eng.Radityo Anggoro, S.Kom, M.Sc

Dosen Pembimbing II
Dimas Anton Asfani S.T, M.T, Ph.D

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR - KI141502

Rancang Bangun Sistem Cerdas Berbasis Cloud Pada Studi Kasus Sistem Monitoring Daya *Charging Station* Mobil Listrik

MUHAMMAD FARID WAJDI
NRP 5112100194

Dosen Pembimbing I
Dr.Eng.Radityo Anggoro, S.Kom, M.Sc

Dosen Pembimbing II
Dimas Anton Asfani, S.T, M.T, Ph.D

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESES - KI141502

DESIGN OF INTELLIGENT SYSTEM CLOUD BASED CASE STUDY MONITORING SYSTEM POWER AN ELECTRIC CAR CHARGING STATION

**MUHAMMAD FARID WAJDI
NRP 5112100194**

**Supervisor I
Dr.Eng.Radityo Anggoro, S.Kom, M.Sc**

**Supervisor II
Dimas Anton Asfani, S.T, M.T, Ph.D**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Rancang Bangun Sistem Cerdas Berbasis Cloud Pada Studi Kasus Sistem Monitoring Daya Charging Station Mobil Listrik

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

MUHAMMAD FARID WAJDI

NRP : 5112 100 194

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Dr.Eng.Radityo Anggoro, S.Kom, M.Ts, Ph.D.
NIP. 198410162008121002 (Pembimbing 1)
2. Dimas Anton Asfani S.T, M.Ts, Ph.D.
NIP: 198109052005011002 (Pembimbing 2)

**SURABAYA
JUNI, 2016**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN SISTEM CERDAS BERBASIS CLOUD PADA STUDI KASUS SISTEM MONITORING DAYA CHARGING STATION MOBIL LISTRIK

Nama Mahasiswa : MUHAMMAD FARID WAJDI
NRP : 511210014
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr.Eng.Radityo Anggoro, S.Kom, M.Sc
Dosen Pembimbing 2 : Dimas Anton Asfani S.T, M.T, Ph.D

Abstrak

Seiring dengan berkembangnya kendaraan listrik, dibutuhkan stasiun pengisian energi listrik (Electric Vehicle Charging Station) yang dilengkapi dengan infrastruktur yang baik. Pada stasiun pengisian tersebut, terdapat berbagai parameter pengukuran yang perlu dimonitor, seperti arus dan tegangan. Oleh karena itu, dalam tugas akhir ini didesain sebuah sistem monitoring dan kontrol berbasis cloud untuk memantau dan mengontrol kinerja sistem pada Charging Station ITS. Sensor untuk mengukur tegangan AC pada sistem adalah ZMPT101B. Untuk mengukur arus AC pada saluran pengisian (Torch) dan output inverter menggunakan sensor LA 55-P. Sedangkan untuk mengukur tegangan DC pada PV dan baterai menggunakan rangkaian voltage divider dan untuk mengukur arus DC menggunakan shunt resistor. Pengambilan data arus dan tegangan AC dilakukan dengan sampling sebanyak 300 data dan diambil nilai tertingginya. Sedangkan pengambilan data arus dan tegangan DC dilakukan dengan penjumlahan sampling 20 data yang kemudian di rata-rata. Hasil pembacaan sensor dikirimkan ke server untuk disimpan dalam database dan data terakhir akan ditampilkan pada aplikasi system monitoring. Pada aplikasi monitoring ini terdapat tombol switch ON/OFF untuk memutus

atau menyambung aliran daya pada sistem dengan menggunakan Solid State Relay. Sistem monitoring dan kontrol ini menggunakan pemrograman Arduino IDE. Pengujian dilakukan di Charging Station ITS dengan terhubung web untuk melihat performa sensor. Error sensor teruji tidak lebih dari 5%.

Kata kunci: Cloud, Charging Station, Sistem Monitoring dan Kontrol

DESIGN OF INTELLIGENT SYSTEM CLOUD BASED CASE STUDY MONITORING SYSTEM POWER AN ELECTRIC CAR CHARGING STATION

Student's Name : MUHAMMAD FARID WAJDI
Student's ID : 5112100194
Department : Teknik Informatika FTIF-ITS
First Advisor : Dr.Eng Radityo Anggoro S.kom, M.sc
Second Advisor : Dimas Anton Asfani, S.T, M.T, Ph.D

Abstract

Along with the development of electric vehicles, electrical energy charging stations required (Electric Vehicle Charging Station) equipped with good infrastructure. At the filling station, there are a variety of measurement parameters to be monitored, such as current and voltage. Therefore, in this thesis designed a monitoring and control system that is tele-monitoring to monitor and control the performance of the system at Charging Station ITS. Sensors to measure the AC voltage on the system is ZMPT101B. To measure the AC current in the drain plug (Torch) and the output inverter using sensor LA 55-P. Meanwhile, to measure DC voltage on the PV and battery using a voltage divider circuit and to measure DC current using a shunt resistor. Data retrieval of AC current and voltage is done by sampling as many as 300 data and taken its highest value. While the data collection of DC current and voltage is done by sampling 20 data summation later on average. The results of sensor readings are sent to the server to be stored in a MySQL database and the latest data will be displayed on the web interface server. In this web interface there is a button switch ON / OFF to connect or disconnect the power flow on the system by using a Solid State Relay. Monitoring and control system using the Arduino programming IDE. Tests carried out at Charging Station

ITS to connect the web to see the performance of the sensor. Error sensor is tested not more than 5%.

Keyword: Cloud, Charging Station, Monitoring and Control System

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alam, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“RANCANG BANGUN SISTEM CERDAS BERBASIS CLOUD PADA STUDI KASUS SISTEM MONITORING DAYA CHARGING STATION MOBIL LISTRIK”**. Shalawat serta salam selalu senantiasa saya tujukan kepada Rasulullah SAW, selaku inspirator dunia nomor satu saat ini, hingga akhir zaman.

Pengerjaan Tugas Akhir ini merupakan salah satu dari sekian banyak kesempatan yang saya dapatkan, untuk mendapatkan ilmu dan pengalaman berharga selama saya berada di kampus Teknik Informatika ITS ini. Dengan pengerjaan Tugas Akhir ini, saya menjadi semakin bisa untuk manajemen waktu dan manajemen diri sendiri, sehingga Tugas Akhir ini dapat selesai tepat waktu.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Ibu, Ibu, Ibu, Ayah dan Kakak yang selalu memberikan do'a, dukungan, serta motivasi, sehingga penulis selalu termotivasi untuk menyelesaikan Tugas Akhir.
3. Bapak Radityo Anggoro, S.kom, M.sc, selaku pembimbing I yang selalu menyemangati dan memotivasi dengan ilmu-ilmu yang diluar dugaan saya.
4. Bapak Dimas Anton Asfani Ph.D selaku pembimbing II yang telah mengoreksi buku ini dengan cermat.
5. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA yang selalu meluangkan

waktunya untuk *sharing* ilmu bersama saya, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.

6. Kepada Shofi Dwika Adha yang telah memberikan doa, memberikan saya semangat untuk mengerjakan Tugas Akhir ini dan telah membantu merapikan buku Tugas Akhir ini.
7. Teman – Teman dalam proyek charging station diantaranya Ubaidillah Husni , Ruli Huda dan Sanindya Lesario.
8. Teman – teman TC Hura Hura, yang telah menemani penulis untuk berpetualang bersama mengarungi pulau Jawa, sampai saat ini.
9. Teman – teman angkatan 2012, tanpa mereka, saya tidak akan merasakan apa itu yang dinamakan “Angkatan”.
10. Teman – teman administrator Laboratorium Komputasi Berbasis Jaringan, yang telah memberikan saya kesempatan untuk mempelajari lebih dalam tentang KBJ, serta teman lab lain yang telah mengajari saya banyak sekali, iya, banyak sekali hal secara tidak langsung.
11. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2016

DAFTAR ISI

LEMBAR PENGESAHAN	v
<i>Abstrak</i>	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xvii
DAFTAR KODE SUMBER.....	xix
BAB I PENDAHULUAN	Error! Bookmark not defined.
1.1 Latar Belakang.....	xvii
1.2 Rumusan Masalah.....	Error! Bookmark not defined.
1.3 Batasan Masalah	Error! Bookmark not defined.
1.4 Tujuan.....	Error! Bookmark not defined.
1.5 Manfaat.....	Error! Bookmark not defined.
1.6 Metodologi.....	Error! Bookmark not defined.
1.7 Sistematika Penulisan Laporan Tugas Akhir	Error! Bookmark not defined.
BAB II TINJAUAN PUSTAKA	Error! Bookmark not defined.
2.1 <i>Charging Station</i>	Error! Bookmark not defined.
2.2 <i>Mikrokontroler Arduino</i>	Error! Bookmark not defined.
2.3 <i>Ethernet Shield</i>	Error! Bookmark not defined.
2.4 Sensor Arus <i>LA 55-P</i>	Error! Bookmark not defined.
2.5 Sensor Tegangan <i>ZMPT101B</i>	Error! Bookmark not defined.
2.6 <i>Shunt Resistor</i>	Error! Bookmark not defined.
2.7 Rangkaian <i>Voltage Divider</i> ..	Error! Bookmark not defined.
2.8 <i>Solid State Relay</i>	Error! Bookmark not defined.
2.9 <i>Router</i>	Error! Bookmark not defined.
2.10 <i>PHP</i>	Error! Bookmark not defined.
2.11 <i>Cloud</i>	Error! Bookmark not defined.
BAB III DESAIN PERANGKAT.....	Error! Bookmark not defined.

3.1	Deskripsi Umum Sistem	Error! Bookmark not defined.
3.2	Perancangan Perangkat Keras.....	Error! Bookmark not defined.
3.2.1	Rangkaian <i>Power Supply</i>	Error! Bookmark not defined.
3.2.2	Rangkaian Sensor Arus AC LA 55-P...	Error! Bookmark not defined.
3.2.3	Rangkaian Sensor Tegangan AC ZMPT101B.....	Error! Bookmark not defined.
3.2.4	Rangkaian <i>Solid State Relay</i>	Error! Bookmark not defined.
3.3	Perancangan Aplikasi Monitoring	Error! Bookmark not defined.
3.3.1	Perancangan <i>Cloud Computing</i>	Error! Bookmark not defined.
3.3.2	Perancangan Metode Pembacaan Nilai Arus dan Tegangan DC.....	Error! Bookmark not defined.
3.3.3	Perancangan Metode Pembacaan Nilai Arus dan Tegangan AC.....	Error! Bookmark not defined.
3.3.4	Perancangan Diagram <i>Use Case</i> ...	Error! Bookmark not defined.
3.3.5	Perancangan Basis Data..	Error! Bookmark not defined.
3.3.6	Perancangan <i>Web Service</i>	Error! Bookmark not defined.
3.3.7	Perancangan Antarmuka Sistem...	Error! Bookmark not defined.
BAB IV IMPLEMENTASI.....		Error! Bookmark not defined.
4.1	Lingkungan Implementasi ...	Error! Bookmark not defined.
4.2	Implementasi	Error! Bookmark not defined.
4.2.1	Implementasi Mikrokontroler dan Komponen Sensor	Error! Bookmark not defined.
4.2.2	Implementasi Aplikasi Monitoring..	Error! Bookmark not defined.
BAB V UJI COBA DAN EVALUASI		Error! Bookmark not defined.
5.1	Lingkungan Uji Coba	Error! Bookmark not defined.
5.2	Skenario Uji Coba	Error! Bookmark not defined.

BAB VI KESIMPULAN DAN SARAN.....	21
6.1 Kesimpulan.....	21
6.2 Saran	21
DAFTAR PUSTAKA.....	73
LAMPIRAN	75
BIODATA PENULIS	77

DAFTAR GAMBAR

Gambar 2.1	Skema <i>hybrid charging station</i> ITS	8
Gambar 2.2	Bentuk fisik (a) konektor <i>SAE J1772</i> dan (b) <i>charging transaction</i>	9
Gambar 2.3	<i>Mikrokontroler Arduino</i>	10
Gambar 2.4	<i>Arduino Uno</i>	11
Gambar 2.5	<i>Arduino Due</i>	11
Gambar 2.6	<i>Arduino Leonardo</i>	12
Gambar 2.7	<i>Arduino Fio</i>	12
Gambar 2.8	<i>Arduino Mega</i>	13
Gambar 2.9	<i>Ethernet Shield</i>	14
Gambar 2.10	Sensor Arus <i>LA 55-P</i>	15
Gambar 2.11	Rangkaian pengukuran dasar <i>LA 55-P</i>	16
Gambar 2.12	Sensor Tegangan <i>ZMPT101B</i>	16
Gambar 2.13	<i>Shunt Resistor</i>	17
Gambar 2.14	Rangkaian <i>Voltage Divider</i>	18
Gambar 2.15	Bentuk Fisisk <i>Solid State Relay</i>	19
Gambar 2.16	Rangkaian <i>Solid State Relay</i>	22
Gambar 2.17	Bentuk <i>TP-LINK TL MR3420</i>	23
Gambar 2.18	Sistem <i>Cloud Computing</i>	25
Gambar 2.19	Heroku Flow Chart.....	26
Gambar 3.1	Diagram blok sistem monitoring dan kontrol <i>Charging Station</i>	28
Gambar 3.2	Skema perancangan perangkat keras.....	30
Gambar 3.3	Perancangan perangkat keras	30
Gambar 3.4	Skematik rangkaian <i>power supply</i>	30
Gambar 3.5	Skematik rangkaian sensor arus <i>LA 55-P</i>	31
Gambar 3.6	Rangkaian <i>ZMPT101B</i>	32
Gambar 3.7	Rangkaian <i>solid state relay</i>	33
Gambar 3.8	Flowchart metode pembacaan nilai arus dan tegangan DC.....	35
Gambar 3.9	Flowchart metode pembacaan nilai arus dan tegangan AC.....	36
Gambar 3.10	<i>use case diagram</i>	37

Gambar 3.11	Tampilan dashboard web <i>Charging Station</i> ITS	41
Gambar 3.12	Tampilan database web <i>Charging Station</i> ITS	41
Gambar 3.13	Tampilan tab grafik pada aplikasi monitoring <i>charging station</i> ITS	42
Gambar 3.14	Tampilan tab kontrol pada aplikasi monitoring <i>charging station</i> ITS	43
Gambar 4.1	Halaman sistem monitoring <i>Charging Station</i>	52
Gambar 4.2	Halaman history data charging station	54
Gambar 4.3	Halaman grafik <i>real time</i> charging station	56
Gambar 5.1	Pemasangan LA 55-P pada kabel <i>Charging Station</i>	58
Gambar 5.2	Grafik hasil pengujian nilai arus terukur pada inverter	59
Gambar 5.3	Pemasangan <i>ZMPT101B</i> pada bus AC <i>Charging Station</i>	60
Gambar 5.4	Grafik hasil pengujian nilai tegangan terukur pada bus AC	61
Gambar 5.5	Pengujian control relay via web saat ON dan OFF	62
Gambar 5.6	Tampilan database web <i>Charging Station</i> ITS	63
Gambar 5.7	Tampilan data dashboard web <i>Charging Station</i> ITS	63
Gambar 5.8	Tampilan <i>flow diagram</i> web <i>Charging Station</i> ITS	64
Gambar A.1	Uji coba linearisasi sensor arus LA 55-P	75
Gambar A.2	Uji coba linearisasi sensor tegangan <i>ZMPT101</i>	76

1.1 Latar Belakang

Ketergantungan terhadap bahan bakar fosil sebagai sumber energi utama membawa kita pada krisis energi dan berbagai masalah lingkungan. Penggunaan bahan bakar fosil termasuk salah satu penyumbang masalah lingkungan terbesar seperti global warming dan hujan asam, akibat emisi gas yang dibuang ke lingkungan. Berbagai usaha yang berkaitan dengan pengurangan jumlah konsumsi energi fosil mulai bermunculan. Dalam sektor transportasi, kendaraan listrik telah menjadi solusi yang menjanjikan dalam usaha pengurangan dalam pemakaian energi fosil, dan kendaraan listrik terus berkembang menjadi pasar utama dalam industri otomotif.

Seiring dengan berkembangnya kendaraan listrik, dibutuhkan stasiun pengisian energi listrik (*Charging Station*) yang dilengkapi dengan infrastruktur yang baik. Dalam stasiun pengisian tersebut, terdapat berbagai parameter pengukuran yang perlu dimonitor [1]. Pada umumnya stasiun pengisian energi listrik tersebut menggunakan sumber *hybrid* dari grid PLN dan PV (*Photovoltaic*). PV menjadi energi alternatif yang menjanjikan karena potensi energi matahari di Indonesia yang sangat besar serta memiliki banyak kelebihan seperti bebas polusi dan mudah diaplikasikan.

Kendaraan listrik merupakan beban non-linear yang dapat menimbulkan harmonisa ketika dihubungkan dengan sumber, sehingga mempengaruhi kualitas daya sistem [2]. Dengan melihat kondisi kerja yang kompleks tersebut, diperlukan sebuah sistem khusus untuk memantau dan mengontrol kondisi kerja. Monitoring arus dan tegangan dilakukan pada setiap titik pada *Charging Station* seperti PV, baterai, *inverter*, maupun beban untuk mengetahui kondisi terkini. Sedangkan kontroling menggunakan relay sebagai *switch* ON/OFF pada saluran (*torch*) dan sumber pengisian dilakukan ketika ada gangguan ataupun saat *maintenance* agar tidak

merusak keseluruhan sistem. Sistem ini diharapkan dapat memberikan layanan yang aman dan handal dalam pengoperasian kendaraan listrik.

Cloud Computing merupakan kemajuan teknologi dalam pengiriman informasi dan jasa. Dengan menyediakan akses sesuai kebutuhan pada *Resources Computing* yang telah di kumpulkan dan diskalakan secara dinamis menurut pengukurannya, *Cloud Computing* menawarkan keuntungan yang menarik dalam hal kecepatan, kelincahan dan efisiensi. Melalui teknologi *Cloud Computing* ini, dapat dibangun sebuah sistem pintar yang dapat melakukan monitoring kondisi arus listrik yang mengalir dari *Charging Station* ke mobil listrik yang ingin melakukan isi ulang secara real time [3].

Agar proses monitoring dan kontrol dapat dilakukan dari jarak jauh tanpa harus mendatangi *plant*, maka diperlukan sebuah sistem monitoring yang dapat mempermudah admin dalam melakukan monitoring *charging station* terhadap mobil listrik. Oleh karena itu pada tugas akhir ini dilakukan rancangan sebuah sistem monitoring dan kontrol arus dan tegangan pada stasiun pengisian kendaraan listrik berbasis cloud. Dengan menggunakan *cloud server*, maka *client* dapat dengan mudah memonitoring arus dan tegangan pada setiap titik di *Charging Station* secara *real time*.

DAFTAR TABEL

Tabel 3.1	Perancangan Tabel <i>user</i>	38
Tabel 3.2	Perancangan Tabel <i>tb_charging</i>	38
Tabel 5.1	Hasil pengujian rangkaian sensor arus LA 55-P (lampu depan)	55
Tabel 5.2	Hasil pengujian rangkaian sensor arus LA 55-P (inverter)	56
Tabel 5.3	Hasil pengujian rangkaian sensor arus LA 55-P (kontainer)	58
Tabel 5.4	Hasil pengujian rangkaian sensor arus LA 55-P (Torch 1).....	59

Tabel 5.5 Hasil pengujian rangkaian sensor arus LA 55-P (Torch 2)..... 60

Tabel 5.6 Hasil pengujian sensor tegangan ZMPT101B 62

Tabel 5.7 Hasil pengujian nilai arus terukur pada inverter..... 64

Tabel 5.8 Hasil pengujian nilai tegangan terukur pada bus AC 66

Tabel A.1 Uji coba linearisasis sensor arus *LA 55-P* 75

Tabel A.2 Uji coba linearisasi sesnsor tegangan *ZMPT101B* . 76

[Halaman ini sengaja dikosongkan

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Pembacaan Nilai Arus dan tegangan AC	46
Kode Sumber 4.2 Implementasi Pembacaan Nilai Arus dan tegangan DC	47
Kode Sumber 4.3 Implementasi <i>Web Service</i> pada Arduino.....	48
Kode Sumber 4.4 Implementasi kontrol pada Arduino	49
Kode Sumber 4.5 Implementasi Perancangan <i>Cloud Heroku</i> ...	50
Kode Sumber 4.6 Implementasi Pengiriman Data ke Database	51
Kode Sumber 4.7 Implementasi Monitoring Real Time	51
Kode Sumber 4.8 Implementasi Melihat History Data	53
Kode Sumber 4.9 Implementasi Grafik JSON	54
Kode Sumber 4.10 Implementasi grafik pada aplikasi monitoring	54

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ketergantungan terhadap bahan bakar fosil sebagai sumber energi utama membawa kita pada krisis energi dan berbagai masalah lingkungan. Penggunaan bahan bakar fosil termasuk salah satu penyumbang masalah lingkungan terbesar seperti global warming dan hujan asam, akibat emisi gas yang dibuang ke lingkungan. Berbagai usaha yang berkaitan dengan pengurangan jumlah konsumsi energi fosil mulai bermunculan. Dalam sektor transportasi, kendaraan listrik telah menjadi solusi yang menjanjikan dalam usaha pengurangan dalam pemakaian energi fosil, dan kendaraan listrik terus berkembang menjadi pasar utama dalam industri otomotif.

Seiring dengan berkembangnya kendaraan listrik, dibutuhkan stasiun pengisian energi listrik (*Charging Station*) yang dilengkapi dengan infrastruktur yang baik. Dalam stasiun pengisian tersebut, terdapat berbagai parameter pengukuran yang perlu dimonitor [1]. Pada umumnya stasiun pengisian energi listrik tersebut menggunakan sumber *hybrid* dari grid PLN dan PV (*Photovoltaic*). PV menjadi energi alternatif yang menjanjikan karena potensi energi matahari di Indonesia yang sangat besar serta memiliki banyak kelebihan seperti bebas polusi dan mudah diaplikasikan.

Kendaraan listrik merupakan beban non-linear yang dapat menimbulkan harmonisa ketika dihubungkan dengan sumber, sehingga mempengaruhi kualitas daya sistem [2]. Dengan melihat kondisi kerja yang kompleks tersebut, diperlukan sebuah sistem khusus untuk memantau dan mengontrol kondisi kerja. Monitoring arus dan tegangan dilakukan pada setiap titik pada *Charging Station* seperti PV, baterai, *inverter*, maupun beban untuk mengetahui kondisi terkini. Sedangkan kontroling menggunakan relay sebagai *switch* ON/OFF pada saluran (*torch*) dan sumber pengisian dilakukan ketika

ada gangguan ataupun saat *maintenance* agar tidak merusak keseluruhan sistem. Sistem ini diharapkan dapat memberikan layanan yang aman dan handal dalam pengoperasian kendaraan listrik.

Cloud Computing merupakan kemajuan teknologi dalam pengiriman informasi dan jasa. Dengan menyediakan akses sesuai kebutuhan pada *Resources Computing* yang telah di kumpulkan dan diskalakan secara dinamis menurut pengukurannya, *Cloud Computing* menawarkan keuntungan yang menarik dalam hal kecepatan, kelincihan dan efisiensi. Melalui teknologi *Cloud Computing* ini, dapat dibangun sebuah sistem pintar yang dapat melakukan monitoring kondisi arus listrik yang mengalir dari *Charging Station* ke mobil listrik yang ingin melakukan isi ulang secara real time [3].

Agar proses monitoring dan kontrol dapat dilakukan dari jarak jauh tanpa harus mendatangi *plant*, maka diperlukan sebuah sistem monitoring yang dapat mempermudah admin dalam melakukan monitoring *charging station* terhadap mobil listrik. Oleh karena itu pada tugas akhir ini dilakukan rancangan sebuah sistem monitoring dan kontrol arus dan tegangan pada stasiun pengisian kendaraan listrik berbasis cloud. Dengan menggunakan *cloud server*, maka *client* dapat dengan mudah memonitoring arus dan tegangan pada setiap titik di *Charging Station* secara *real time*.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana implementasi sistem monitoring daya *Charging Station* mobil listrik terhadap tampilan data pada sistem cerdas berbasis *cloud*?
2. Bagaimana memproses data analog dari sensor agar menjadi data digital yang merepresentasikan nilai sebenarnya dari arus dan tegangan yang diukur?
3. Bagaimana implementasi sistem monitoring *Charging Station* dengan melakukan kontrol relay sebagai *switch* ON/OFF?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Sistem monitoring daya *Charging Station* menggunakan bahasa pemrograman PHP dan *database* MySQL.
2. Sensor arus AC yang digunakan adalah *LEM LA 55-P*.
3. Sensor tegangan AC yang digunakan adalah *ZMPT101B*.
4. Kontrol switch ON/OFF menggunakan *Solid State Relay* (SSR).
5. Mikrokontroler yang digunakan ialah *Arduino Mega 2560* dengan *Ethernet shield* sebagai modul untuk terhubung dengan jaringan.
6. Router yang digunakan untuk membuat jaringan lokal ialah *TP-LINK*.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini antara lain :

1. Mengimplementasikan sistem monitoring daya *Charging Station* pada Sistem Cerdas secara *real time*.
2. Menentukan teknik yang tepat untuk melakukan pembacaan nilai sebenarnya dari arus dan tegangan.
3. Mewujudkan sebuah sistem monitoring dengan menggunakan kontrol secara *real time*.

1.5 Manfaat

Hasil dari tugas akhir ini diharapkan mampu memudahkan *user/admin charging station* dalam melakukan monitoring dan kontroling parameter-parameter pengukuran pada berbagai titik di stasiun pengisian kendaraan listrik secara *realtime*, dapat membantu menghitung keluaran arus yang keluar dari masing objek pengukuran yang dapat bermanfaat untuk menghitung besaran biaya yang diperlukan setiap melakukan pengisian mobil listrik pada *Charging Station*.

1.6 Metodologi

1. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan pada proposal tugas akhir ini terdiri dari latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat hasil dari pembuatan tugas akhir. Selain itu dijelaskan pula tinjauan pustaka yang digunakan sebagai referensi pendukung implementasi tugas akhir. Pada proposal ini juga terdapat perencanaan jadwal pengerjaan tugas akhir.

2. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam implementasi sistem, yaitu mengenai mikrokontroler Arduino, menampilkan data secara *real time*, pemilihan sensor arus dan tegangan rating tinggi.

3. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sitem berdasarkan studi literature dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat prototype sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

4. Implementasi perangkat lunak

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

5. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba pada alat yang telah dirancang. Tahapan ini dimaksudkan untuk mengevaluasi tingkat akurasi dari alat tersebut serta mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

6. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Perancangan Perangkat Lunak

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *flow chart*.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa code yang digunakan untuk proses implementasi.

Bab V Uji Coba Dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan Dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan rancangan alat yang diajukan pada pengimplementasian program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap alat yang dirancang dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

1.1 *Charging Station*

Stasiun pengisian kendaraan listrik atau yang biasa disebut *Electric Vehicle Charging Station* adalah sumber eksternal untuk pengisian kendaraan listrik. Setiap stasiun dilengkapi dengan *plug* yang dihubungkan ke kendaraan. Stasiun pengisian (*charging station*) dapat memiliki lebih dari satu sumber energi listriknya. Gambar 2.1 menunjukkan skema dari stasiun pengisian kendaraan listrik di ITS. Dalam skema terlihat bahwa *charging station* ITS merupakan *hybrid charging station* karena sumber energi listriknya merupakan integrasi dari *grid* (PLN), Genset, dan *photovoltaic* (PV) yang dilengkapi dengan media penyimpanan berupa baterai. Terdapat tiga slot untuk pengisian kendaraan listrik yaitu dua untuk mobil listrik dan satu untuk bus listrik [4]. Untuk karakteristik pengisiannya adalah sebagai berikut:

- Mobil Listrik
 - *Slow Charging*

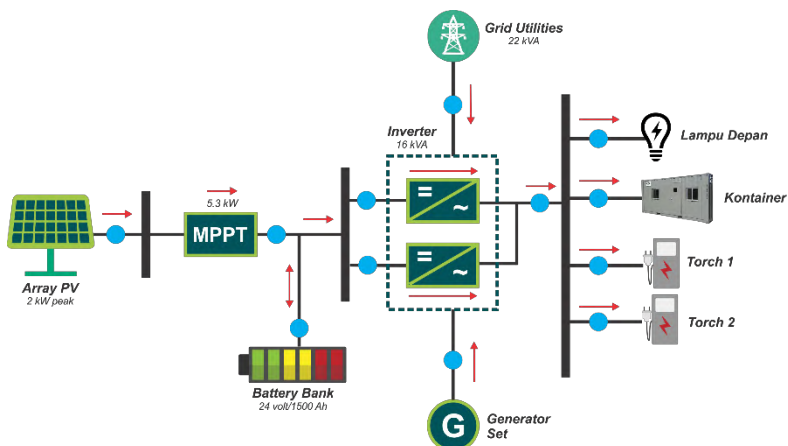
Arus pengisian	: 6 A @ 220 VAC
Kapasitas daya	: 1.320 Watt
Waktu pengisian	: 10 jam
 - *Fast Charging*

Arus pengisian	: 11 A @ 220 VAC
Kapasitas daya	: 2.420 Watt
Waktu pengisian	: 5-6 jam

- Bus Listrik

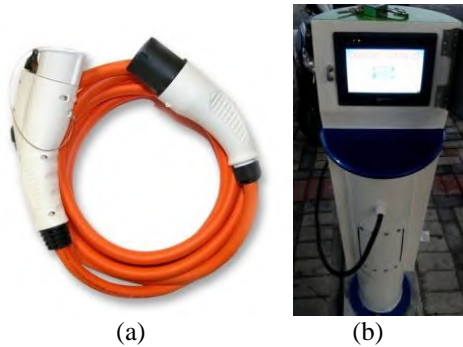
Arus Pengisian	: 50 A @ 220 VAC
Kapasitas daya	: 11.000 Watt
Waktu pengisian	: 10 jam

Photovoltaic sebagai sumber energi terbarukan yang dipakai dalam sistem terdiri dari 10 modul dengan rating 2.000 Watt *peak*, sedangkan baterai sebagai media penyimpan terdiri dari 12 buah baterai 2 Volt dengan rating kapasitas total sebesar 1.500 Ah.



Gambar 2.1 Skema hybrid charging station ITS

Charging Station ITS termasuk dalam *AC Charging Station* sehingga dalam sistem ini, output yang dihasilkan untuk pengisian mobil atau bus adalah arus AC. PV dan baterai selanjutnya akan masuk ke inverter bersama sumber dari *grid* untuk kemudian disalurkan ke beban. *Charging station ITS* akan menggunakan standar *SAE J1772* dan *charging transaction* seperti pada Gambar 2.2.



Gambar 2.2 Bentuk fisik (a) konektor SAE J1772 dan (b) *charging transaction* [4]

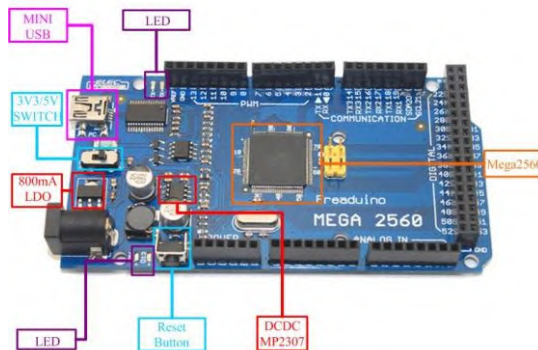
1.2 Mikrokontroler Arduino

Arduino sebagai mikro single-board yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki *processor Atmel AVR* dan softwarenya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia [5]. Bahasa yang dipakai dalam Arduino bukan assembler yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler, sekaligus menawarkan berbagai macam kelebihan antara lain:

1. Perangkat lunaknya Open Source – Perangkat lunak Arduino IDE dipublikasikan sebagai Open Source, tersedia bagi para pemrogram berpengalaman untuk pengembangan lebih lanjut. Bahasanya bisa dikembangkan lebih lanjut melalui pustaka-pustaka C++ yang berbasis pada Bahasa C untuk AVR.
2. Perangkat kerasnya Open Source – Perangkat keras Arduino berbasis mikrokontroler *ATMEGA8*, *ATMEGA168*, *ATMEGA328* dan *ATMEGA1280* (yang terbaru

- ATMEGA2560*). Dengan demikian bisa membuat perangkat keras Arduino ini dan bootloader tersedia langsung dari perangkat lunak Arduino IDE-nya. Bisa menggunakan breadboard untuk membuat perangkat Arduino beserta periferal-periferal lain yang dibutuhkan.
3. Tidak perlu perangkat chip programmer karena di dalamnya sudah ada bootloader yang akan menangani upload program dari komputer.
 4. Memiliki sarana komunikasi USB, sehingga pengguna Laptop yang tidak memiliki port serial/RS323 bisa menggunakannya.
 5. Bahasa pemrograman relatif mudah karena software Arduino dilengkapi dengan kumpulan library yang cukup lengkap.
 6. Memiliki modul siap pakai (shield) yang bisa ditancapkan pada board Arduino. Misalnya shield GPS, Ethernet, SD Card, dll.

Tampilan dan konfigurasi pin Arduino Mega 2560 adalah seperti Gambar 2.10 di bawah ini:



Gambar 2.10 Arduino Mega 2560 [5]

Spesifikasi Arduino Mega2560 adalah sebagai berikut:

- Mikrokontroler = ATmega2560
- Flash Memory = 2560 K Byte
- SRAM = 8K Byte

- EEPROM = 4 KB
- Clock Speed = 16Mhz
- DC arus untuk 3,3 v = 50mA
- DC arus untuk i/o = 40mA
- Analog input pin = 40
- Digital I/O pin = 54
- Tegangan Input = 6 – 20 Volt
- Tegangan kerja = 5 Volt

Pada Sistem Aplikasi Monitoring ini menggunakan Arduino Mega 2560 karena Arduino tersebut memiliki 54 buah *Digital I/O* pin dibandingkan dengan Arduino UNO R3 yang hanya memiliki 14 buah *Digital I/O* pin. Sistem aplikasi Monitoring harus menggunakan banyak pin sebagai tempat-tempat sensor yang akan dihubungkan ke Arduino.

1.3 *Ethernet Shield*

Ethernet Shield adalah modul yang dapat dipasangkan langsung di atas *board* Arduino untuk menambahkan fungsi LAN / Ethernet sehingga memungkinkan Arduino terhubung dengan internet. Dengan menggunakan modul komunikasi ini, Arduino mendukung protokol TCP/IP atau UDP [5].

Dengan memasang *shield* ini di atas *board* Arduino Mega 2560 (ditumpuk) dan menyambungkan dengan kabel RJ45 dari suatu *network*, Arduino siap dikendalikan lewat intranet/internet dan siap untuk pengiriman data ke *database*. Tentunya dengan pemrograman yang telah dilakukan di IDE Arduino dan pemrograman ini terdiri dari 2 jenis bahasa yaitu bahasa C untuk proses yang dilakukan mikrokontroler berkaitan dengan input/output dan bahasa HTML untuk membuat halaman web yang akan diakses. Berikut Gambar 2.9 merupakan bentuk fisik dari *Ethernet Shield*.



Gambar 2.9 Ethernet Shield [5]

Tegangan suplai dari *shield* ini adalah 5V DC yang didapat langsung dari board arduino. Spesifikasi *Ethernet Controllernya* yaitu chip Wiznet W5100 dengan internal buffer 16 Kb, kecepatan koneksi 10 / 100Mb (Fast-Ethernet). *Shield* ini terhubung dengan Arduino melalui port SPI.

Sebagai tambahan sudah tersedia slot micro-SD card yang dapat digunakan untuk menyimpan file yang akan disajikan ke jaringan. MicroSD card reader ini dapat diakses melalui SD Library (juga sudah tersedia di IDE Arduino).

1.4 Sensor Arus LA 55-P

Sensor arus *LA 55-P* adalah sensor yang diproduksi oleh LEM untuk pengukuran arus AC, DC, *pulsed*, dan sebagainya yang dilengkapi pemisah galvanis di antara rangkaian primer dan rangkaian sekunder. *LA 55-P* dipilih karena memiliki akurasi yang bagus, linieritas yang tinggi, serta tahan terhadap interferensi eksternal. Sensor ini bisa digunakan dalam dua posisi yaitu posisi *reverse* atau *forward* [6]. *Reverse* yaitu posisi sensor terbalik dengan arah arus sehingga pembacaan arus sensor juga akan negative, Sedangkan *forward* yaitu output akan bernilai positif karena pemasangan sensor searah dengan arah arus yang mengalir. Output sensor ini berupa arus, sehingga harus ditambahkan resistor agar output berubah menjadi tegangan yang nantinya akan dimasukkan ke mikrokontroler Arduino. Bentuk sensor *LA 55-P* ditampilkan pada Gambar 2.3.



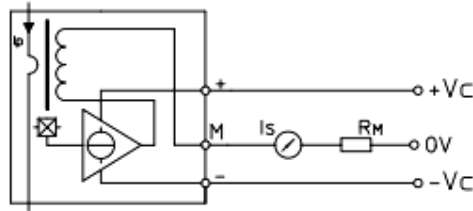
Gambar 2.10 Sensor Arus LA 55-P [6]

Sensor LA 55-P memiliki spesifikasi seperti berikut:

- Current Measuring Range AC: -50A to 50A
- Current Measuring Range DC: -50A to 50A
- Current Range DC: 50mA
- Response Time: 1 μ s
- Sensor Output1: Closed Loop
- Supply Voltage DC Max: 15V
- Supply Voltage DC Min: -15V

Arus nominal yang dapat diukur sensor ini adalah 50 A dengan sumber tegangannya $\pm 12 \dots 15$ V. Sensor arus LA 55-P berdimensi panjang 3,6 cm, lebar 2,7 cm, serta tinggi 1,4 cm. Sedangkan luasan lubang sensor ini yaitu panjang 12,7 mm dan lebar 7 mm. Sensor ini bekerja berdasarkan teori *hall effect* yang sistemnya *closed loop* [7]. Arus diukur dengan cara melewati kabel di lubang yang ada di tengah sensor.

Sensor LA 55-P memiliki 3 pin yang ukurannya standar *pcb mounting* dengan rekomendasi lubang PCB sebesar 1,2 mm yaitu pin +, M, dan -. Gambar 2.4 menunjukkan rangkaian pengukuran dasar sensor LA 55-P. Sesuai dengan *datasheet*, pemasangan R_M bernilai 10 – 100 Ohm pada suhu hingga 70°C, dengan sumber tegangan $+V_C = 12$ V dan $-V_C = -12$ V.



Gambar 2.11 Rangkaian pengukuran dasar LA 55-P [6]

1.5 Sensor Tegangan ZMPT101B

Sensor tegangan ZMPT101B adalah sensor yang biasa digunakan untuk pengukuran tegangan AC satu fasa. Sensor ini mengkonversi tegangan AC input menjadi output tegangan AC yang sudah direduksi tegangannya dengan frekuensi tetap yaitu 50Hz. Menurut datasheet, sensor ini bisa mendeteksi tegangan hingga 1000 volt dengan syarat arusnya tidak boleh lebih dari 2mA, sehingga diharuskan menggunakan resistor sebagai penurun arus [7]. Gambar 2.12 merupakan tampilan dari sensor ZMPT101B.



Gambar 2.12 Sensor Tegangan ZMPT101B [7]

Spesifikasi Sensor ZMPT101B

1. Arus primer = 2mA
2. Arus sekunder = 2mA
3. Turn ratio 1000:1000
4. Phase single error $< 20^\circ$ (input 2mA, sampling resistor 100 ohm)
5. Linear range = 0 – 1000 v
6. Linearity = $< 0.2 \%$
7. Permissible Error = $-0.5\% < f < 0$
8. Isolation voltage = 4000
9. Aplikasi = pengukuran tegangan dan daya

- 10. Encapsulation = Epoxy
- 11. Instalation = PCB mounting
- 12. Operating temperature = $-40 - 70$ derajat celcius

1.6 *Shunt Resistor*

Shunt resistor pada pengukuran suatu rangkaian perlu diletakkan seri terhadap arus yang ingin diukur. Hal ini disebabkan arus tidak akan berubah bila melalui rangkaian seri, dan akan terbagi bila melewati rangkaian yang disusun parallel [8]. Walaupun arus pada rangkaian seri tidak berubah, akan tetapi peletakkan *shunt resistor* pada suatu rangkaian tersebut akan mempengaruhi pengukuran. Hal ini *shunt resistor* sendiri memiliki tahanan internal sehingga akan menambah besaran tahanan total pada rangkaian tersebut, sehingga akan merubah besar arus yang tadinya hanya mengalir ke tahanan pada rangkaian awal. Agar *shunt resistor* tidak mempengaruhi pengukuran maka diperlukan tahanan internal yang kecil. Semakin kecil resistansi dari *shunt resistor*, maka semakin kecil error perhitungan. Kecilnya resistansi juga mempengaruhi sensitivitas alat ukur. Sensitivitas *shunt resistor* adalah jumlah arus yang diperlukan agar terjadi defleksi maksimal pada alat ukur. Semakin kecil jumlah arus, semakin tinggi sensitivitasnya. Berikut Gambar 2.13 menggambarkan bentuk fisik *shunt resistor*.

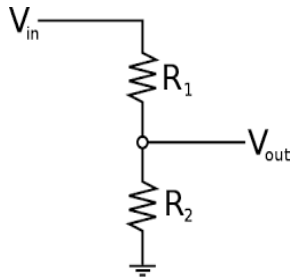


Gambar 2.13 *Shunt Resistor* [8]

1.7 *Rangkaian Voltage Divider*

Rangkaian voltage divider merupakan rangkaian penurun tegangan. Fungsi voltage divider ini adalah untuk mereduksi tegangan

atau menyamakan level tegangan tertinggi input dengan level tegangan baca mikrokontroler dengan range 0 – 5V. Rangkaian voltage divider terdiri dari 2 buah resistor [9]. Seperti pada Gambar 2.14 di bawah ini.



Gambar 2.14 Rangkaian Voltage Divider [9]

1.8 Solid State Relay

Solid state relay (SSR) adalah sebuah saklar elektronik yang tidak memiliki bagian yang bergerak. Contohnya foto-coupled SSR, transformer-coupled SSR, dan hybrida SSR. SSR ini dibangun dengan isolator sebuah MOC untuk memisahkan bagian input dan bagian saklar. Dengan SSR kita dapat menghindari terjadinya percikan api dan sambungan tidak sempurna karena kontaktor keropos seperti yang terjadi pada relay konvensional. SSR menggunakan kontaktor berupa komponen aktif seperti TRIAC, sehingga [relay](#) ini dapat dikendalikan dengan tegangan rendah dan dapat digunakan untuk mengendalikan tegangan AC dengan voltase besar [10]. Contoh fisik dari SSR ditunjukkan pada Gambar 2.15 di bawah ini.



Gambar 2.15 Bentuk fisik *Solid State Relay* (SSR) [10]

Solid state relay dapat diartikan sebagaimana kontrol ON-OFF di mana arus beban dilakukan oleh satu atau lebih semikonduktor, misalnya sebuah transistor daya, sebuah SCR, atau TRIAC.

Penggunaan solid state relay mempunyai beberapa keuntungan yang membuat relay ini banyak digunakan pada aplikasi-aplikasi kontrol untuk beban AC daripada digunakannya relay mekanik (Electromechanical Relay, EMR), walaupun biaya sebuah solid-state relay lebih mahal daripada biaya sebuah relay mekanik biasa. Keuntungan solid-state relay antara lain:

1. Pada solid-state relay tidak terdapat bagian yang bergerak seperti halnya pada relay. Relay mempunyai sebuah bagian yang bergerak yang disebut kontaktor dan bagian ini tidak ada pada solid-state relay. Sehingga tidak mungkin terjadi '*no contact*' karena kontaktor tertutup debu bahkan karat.
2. Tidak terdapat '*bounce*', karena tidak terdapat kontaktor yang bergerak maka pada solid-state relay tidak terjadi peristiwa '*bounce*' yaitu peristiwa terjadinya pantulan kontaktor pada saat terjadi perpindahan keadaan. Dengan kata lain dengan tidak adanya bounce maka tidak terjadi percikan bunga api pada saat kontaktor berubah keadaan.
3. Proses perpindahan dari kondisi 'off' ke kondisi 'on' atau sebaliknya sangat cepat hanya membutuhkan waktu sekitar 10 μ s sehingga solid-state relay dapat dengan mudah

dioperasikan bersama-sama dengan *zero-crossing* detektor. Dengan kata lain opsai kerja solid-state relay dapat disinkronkan dengan kondisi *zero crossing* detektor.

4. Solid-State relay kebal terhadap getaran dan guncangan. Tidak seperti relay mekanik biasa yang kontaktornya dapat dengan mudah berubah bila terkena guncangan/getaran yang cukup kuat pada body relay tersebut.
5. Tidak menghasilkan suara ‘klik’, seperti relay pada saat kontaktor berubah keadaan.
6. Kontaktor output pada solid-state relay secara otomatis ‘*latch*’ sehingga energi yang digunakan untuk aktivasi solid-state relay lebih sedikit jika dibandingkan dengan energi yang digunakan untuk aktivasi sebuah relay. Kondisi ON sebuah solid-state relay akan di-*latch* sampai solid-state relay mendapatkan tegangan sangat rendah, yaitu mendekati nol volt.
7. Solid-State relay sangat sensitif sehingga dapat dioperasikan langsung dengan menggunakan level tegangan CMOS bahkan level tegangan TTL. Rangkaian kontrolnya menjadi sangat sederhana karena tidak memerlukan level konverter.
8. Masih terdapat *couple* kapasitansi antara input dan output tetapi sangat kecil sehingga arus bocor antara input output sangat kecil. Kondisi diperlukan pada peralatan medical yang memerlukan isolasi yang sangat baik.

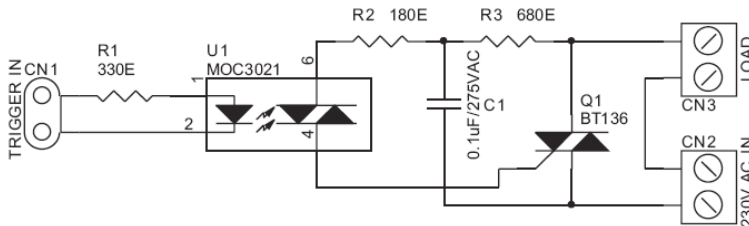
Keuntungan solid-state relay begitu baik sekali tetapi dibalik keuntungan tersebut terdapat kerugian penggunaan solid-state relay yang perlu dipertimbangkan dalam penggunaannya. Kerugian solid-state relay adalah sebagai berikut:

1. Resistansi Tegangan transien. Tegangan yang diatur/dikontrol oleh solid-state relay benar-benar tidak bersih. Dengan kata lain tidak murni tegangannya berupa sinyal sinus dengan tegangan peak to peak 380 vpp tetapi terdapat *spike-spike* yang dihasilkan oleh induksi motor atau peralatan listrik lainnya. Spike ini level tegangannya bervariasi jika terlalu besar maka dapat merusakkan solid-state relay tersebut. Selain itu sumber-

sumber spike yang lain adalah sambaran petir, imbas dari selenoid valve dan lain sebagainya.

2. Tegangan drop. Karena solid-state relay dibangun dari bahan silikon maka terdapat tegangan jatuh antara tegangan input dan tegangan output. Tegangan jatuh tersebut kira-kira sebesar 1 volt. Tegangan jatuh ini menyebabkan adanya dissipasi daya yang besarnya tergantung dari besarnya arus yang lewat pada solid-state relay ini.
3. Arus bocor-‘Leakage current’. Pada saat solid-state relay ini dalam keadaan off atau keadaan open maka dalam kondisi yang idel seharusnya tidak ada arus yang mengalir melewati solid-state relay tetapi tidak demikian pada komponen yang sebenarnya. Besarnya arus bocor cukup besar untuk jika dibandingkan arus pada level TTL yaitu sekitar 10mA rms.
4. Sukar dimplementasikan pada aplikasi multi fasa.
5. Lebih mudah rusak jika terkena radiasi nuklir.

Pada solid state relay, switching unitnya menggunakan TRIAC sehingga relay ini dapat mengalirkan arus baik arus positif maupun arus negatif. Walaupun demikian untuk mengontrol TRIAC ini digunakan SCR yang mempunyai karakteristik gate yang sangat sensitif. Kemudian untuk mengatur trigger pada SCR sendiri diatur dengan menggunakan rangkaian transistor. Rangkaian transistor ini menjadi penguat level tegangan yang didapat dari optocoupler. Penggunaan SCR untuk mengatur gate TRIAC karena gate SCR mempunyai karakteristik yang lebih sensitif daripada gate TRIAC. Skematik rangkaian SSR ditunjukkan pada Gambar 2.16 di bawah ini.



Gambar 2.16 Rangkaian Solid State Relay

1.9 Router

Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya. Router berbeda dengan switch. Switch merupakan penghubung beberapa alat untuk membentuk suatu Local Area Network (LAN). Sebagai ilustrasi perbedaan fungsi dari router dan switch merupakan suatu jalanan, dan router merupakan penghubung antar jalan. Masing-masing rumah berada pada jalan yang memiliki alamat dalam suatu urutan tertentu. Dengan cara yang sama, switch menghubungkan berbagai macam alat, dimana masing-masing alat memiliki alamat IP sendiri pada sebuah LAN [11].

Router sangat banyak digunakan dalam jaringan berbasis teknologi protokol TCP/IP, dan router jenis itu disebut juga dengan IP Router. Selain IP Router, ada lagi AppleTalk Router, dan masih ada beberapa jenis router lainnya. Internet merupakan contoh utama dari sebuah jaringan yang memiliki banyak router IP.

Router yang dipakai pada alat ini adalah TP-LINK TL MR3420 wireless N 150 Mbps seperti pada Gambar 2.17 berikut



Gambar 2.17 Bentuk TP-LINK TL MR3420 [11]

TP-LINK TL MR3420 merupakan jaringan LAN nirkabel berkecepatan tinggi yang telah sesuai dengan standar IEEE 802.11n 150 Mbps serta memiliki fitur High Power. Pada jaringan kabel, LAN ini juga mendukung Gigabit Ethernet yang memiliki kinerja sama walaupun berada pada koneksi kabel atau nirkabel, TP-LINK TL MR3420 membawa era video berkualitas tinggi dan jaringan nirkabel yang ramah lingkungan sehingga memberi kenyamanan tersendiri.

1.10 PHP

PHP (*Personal Home Page*) adalah sebuah bahasa pemrograman yang berjalan pada *webserver* atau *server-side*. PHP dapat digunakan untuk beberapa hal, diantaranya mengolah data dengan tipe apapun, menciptakan halaman web yang dinamis, serta menerima dan menciptakan *cookies*. PHP dapat berjalan pada banyak sistem operasi seperti Linux dan varian Unix (HP-UX, Solaris dan OpenBSD), pada MS Windows, Mac dan lain sebagainya.

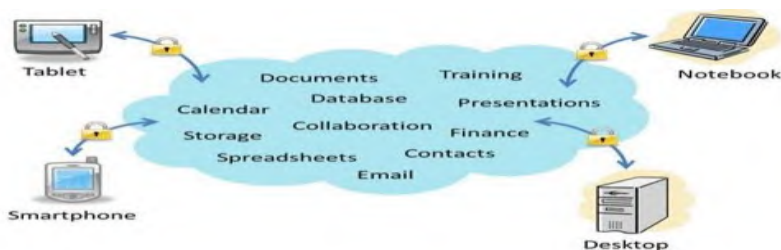
Sistem kerja PHP diawali dengan permintaan halaman *website* oleh sebuah *browser*. Kemudian, ditemukan alamat web *server* dari URL yang dimasukkan pada *browser* dan mengidentifikasi halaman yang dicari dan menyampaikan informasi yang dibutuhkan oleh web *server*. Pada saat permintaan dikirim ke web *server*, web *server* akan memeriksa tipe *file* yang diminta *user*. Jika tipe *file* yang diminta adalah PHP, maka akan memeriksa isi *script* dari halaman PHP tersebut. Apabila dalam *file* tersebut tidak mengandung *script* PHP, permintaan *user* akan langsung ditampilkan ke *browser*. Namun jika

dalam file tersebut mengandung *script* PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan *script* PHP dan mengolah *script* tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke *browser*.

Lalu, *web server* mencari berkas yang diminta dan menampilkan isi berkas pada *browser*. *Browser* yang mendapatkan isi berkas menerjemahkan isi berkas ke dalam bentuk HTML [12]. Bahasa pemrograman PHP ini digunakan pada aplikasi untuk pembuatan *end-user* berupa aplikasi web sebagai hasil akhir dari proses aplikasi ini. Selain itu, bahasa PHP ini digunakan pula pada sisi *server* untuk menangkap *request* data dari aplikasi.

1.11 Cloud

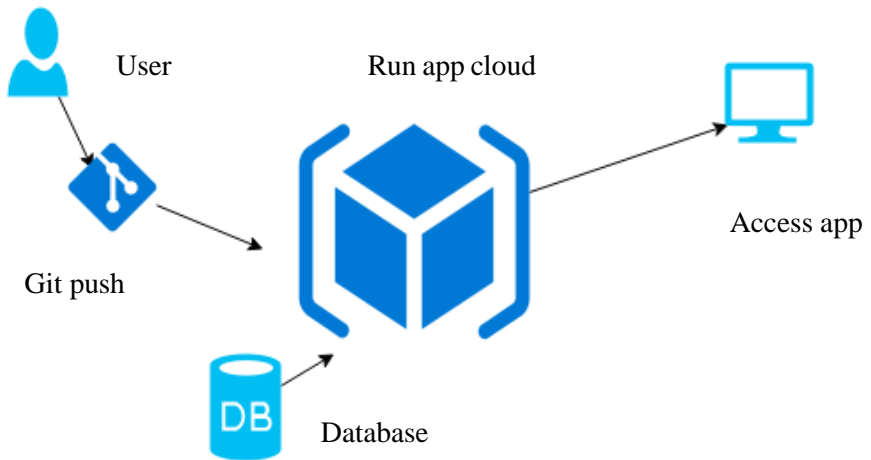
Cloud Computing pada gambar 2.18 adalah sebuah model dimana Resources yang berupa *Processor*, *Memory*, *Storage*, *Network*, dan *Software* menjadi tidak terlihat dan diberikan sebagai layanan di jaringan atau internet menggunakan pola akses remote. teknologi cloud computing ini merupakan sebuah teknologi yang menjadikan internet sebagai pusat server untuk mengelola data dan juga aplikasi pengguna. Teknologi ini mengizinkan para pengguna untuk mengakses data pribadi mereka melalui komputer dengan akses internet. [11]



Gambar 2.18 Sistem Cloud Computing [13]

1.12 Heroku

Heroku adalah sebuah cloud platform yang menjalankan bahasa pemrograman tertentu biasa disebut sebagai platform as a service [14]. Servicenya adalah menjalankan script app langsung tanpa memerlukan setting yang sangat rumit, dengan heroku untuk melakukan deploy cukup melakukan git push dan bisa running secara langsung. Keuntungan menggunakan Heroku selain mudah dan juga tidak memerlukan biaya sewa server. Berikut flow chart Heroku pada Gambar 2.19.



Gambar 2.19 Heroku Flow Chart [14]

[Halaman ini sengaja dikosongkan]

BAB III

DESAIN PERANGKAT

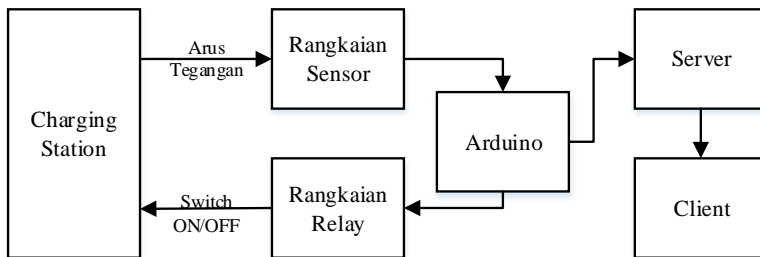
Perancangan merupakan bagian penting dari pembuatan perangkat lunak yang berupa perencanaan-perencanaan secara teknis aplikasi yang dibuat. Sehingga bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir ini. Berawal dari deskripsi umum sistem hingga perancangan perangkat keras dan antar muka sistem.

3.1 Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dibangun suatu sistem monitoring arus daya *Charging Station* mobil listrik dengan menggunakan mikrokontroler *Arduino Mega* berbasis *Cloud*. Aplikasi yang akan dibuat dapat melakukan monitoring kondisi daya listrik yang mengalir dari *Charging Station* ke mobil listrik yang ingin melakukan pengisian ulang. Arus listrik yang mengalir keluar dari *Charging Station* akan dicatat besaran perubahannya dengan menggunakan sensor arus kemudian dari sensor arus tersebut diproses menggunakan mikrokontroler *Arduino* untuk dilakukan pemrosesan data arus listrik yang akan ditampilkan pada website melalui *cloud*. Rangkaian sensor arus dan tegangan serta *relay* dipasang pada titik-titik obyek yang diukur, lalu dihubungkan pada mikrokontroler *Arduino Mega 2560* yang digunakan sebagai pengolah data dan pengontrol *relay*. Rangkaian sensor diberi suplai eksternal dari *Power Supply* sebesar ± 12 V. Data analog hasil pembacaan rangkaian sensor berupa tegangan output yang berkisar 0 – 5 V dihubungkan dengan *Arduino* melalui pin ADC. Data analog tersebut diolah berdasarkan pengujian dan dilinierisasi oleh program *Arduino* untuk mendapatkan hasil pembacaan yang sesuai dengan nilai sebenarnya. Data hasil olahan *Arduino* yang sudah sesuai dengan nilai sebenarnya kemudian dikirimkan ke komputer server dengan jaringan yang sama melalui *Ethernet Shield* secara *realtime*. *Arduino* dipasang modul *Ethernet*

(*Ethernet Shield*) untuk bisa berkomunikasi dan mengirimkan data ke web server. Sebelum ditampilkan pada web, data dari Arduino terlebih dahulu dikirim ke database MySQL dengan menggunakan web service PHP yang terdapat pada aplikasi ini.

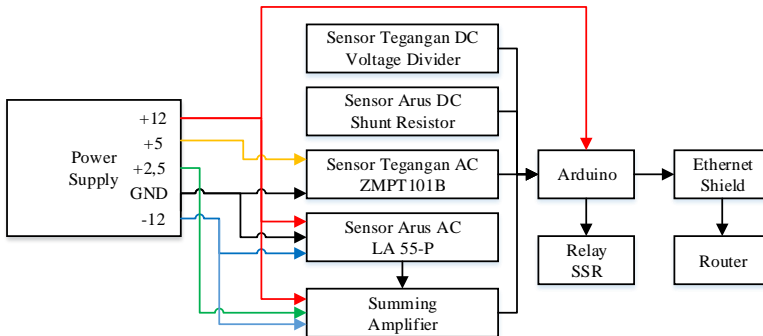
Selain menampilkan data arus dan tegangan yang diukur, *interface* web ini juga memungkinkan untuk melakukan kontrol switch ON/OFF pada relay yang terpasang pada sumber dan *torch*. Rangkaian relay mendapatkan suplai 5V langsung dari Arduino untuk mengaktifkannya. Relay dikontrol melalui pin digital Arduino. Client yang ingin memonitor nilai dari sensor-sensor dan mengontrol relay yang terpasang harus terkoneksi dengan jaringan dan alamat IP yang sama dengan Web Server tersebut, berikut cara kerja sistem monitoring dan control dapat dilihat pada diagram blok gambar 3.1 di bawah ini.



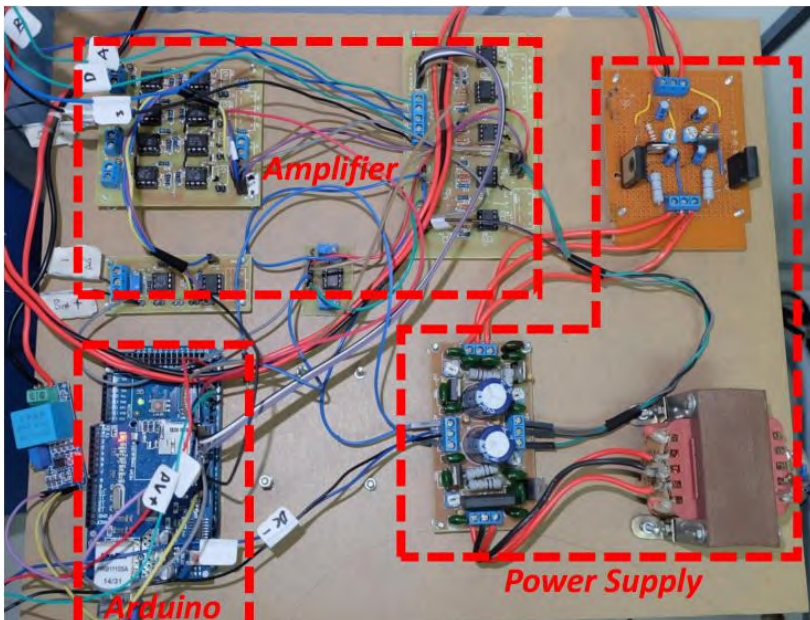
Gambar 3.1 Diagram blok sistem monitoring dan kontrol *chargingstation*

3.2 Perancangan Perangkat Keras

Perancangan perangkat keras untuk sistem monitoring dan kontrol charging station meliputi beberapa komponen utama yaitu rangkaian *Power Supply*, rangkaian sensor, rangkaian relay, mikrokontroler, modul ethernet shield dan router. Berikut Gambar 3.2 merupakan perancangan perangkat keras yang digunakan.



Gambar 3.2 Skema perancangan perangkat keras



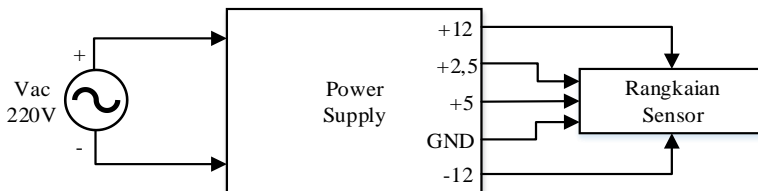
Gambar 3.3 Perancangan perangkat keras

Rangkaian sensor meliputi rangkaian sensor arus AC, sensor tegangan AC, sensor arus DC, dan sensor tegangan DC. Sensor arus AC menggunakan sensor LA 55-P. Sensor ini membutuhkan suplai

+12V, -12V dan GND. Output dari sensor ini berupa tegangan AC dengan skala kecil yaitu 1 : 1000 dari nilai sebenarnya. Sensor tegangan AC menggunakan ZMPT101B. Sensor ini membutuhkan suplai 5V. Sensor arus DC menggunakan Shunt Resistor yang dipasang serial dengan objek yang diukur. Sehingga tidak membutuhkan suplai tegangan dari luar. Sensor tegangan DC menggunakan rangkaian pembagi tegangan (*voltage divider*), rangkaian ini hanya memerlukan input GND dari *Power Supply* karena VCC sudah diperoleh dari tegangan DC yang diukur. Rangkaian relay menggunakan relay SSR yang membutuhkan input tegangan 3-30V DC, sehingga output dari pin digital Arduino sudah cukup untuk digunakan sebagai driver relay ini. Sedangkan Mikrokontroler yang sudah terhubung dengan Ethernet Shield memerlukan suplai tegangan +5V dari USB PC atau +12V dan GND dari *Power Supply*, dan untuk router memiliki *Power Supply* sendiri sebesar +12V.

3.2.1 Rangkaian *Power Supply*

Skematik rangkaian *Power Supply* yang digunakan dapat dilihat pada Gambar 3.4 di bawah ini.



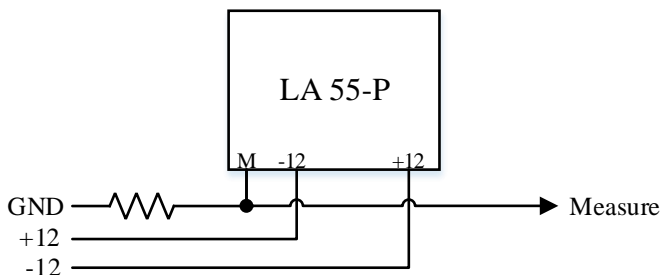
Gambar 3.4 Skematik rangkaian *Power Supply*

Input eksternal dari sumber listrik PLN 220V AC dihubungkan pada trafo *Center Tap* (CT) dan kemudian dipilih output -12...0...12. Output yang masih berupa sinyal AC dengan V_{rms} sekitar 24V AC dimasukkan ke dioda *bridge* untuk disearahkan dan untuk mengurangi *ripple* digunakan kapasitor. Ground diambil dari

titik CT agar mendapat tegangan + dan – kemudian untuk meregulasi tegangan positifnya digunakan IC 7812 agar mendapat tegangan +12V DC dan IC 7912 agar didapat -12V DC.

3.2.2 Rangkaian Sensor Arus AC LA 55-P

Digunakan sensor arus LA 55-P untuk pengukuran arus AC. Sensor ini membutuhkan suplai +12V, -12V dan GND. LA 55-P mempunyai skala 1000 : 1 yang berarti arus input sebesar 1000 A akan terbaca 1 A pada outputnya. Output dari sensor arus LA-55P adalah arus AC yang nantinya akan ditambahkan resistor agar menjadi tegangan. Dibutuhkan rangkaian summing amplifier untuk menjadikan fase negatif tegangan AC bernilai positif dengan menaikkan tegangan offsetnya menjadi 2,5V sebelum dimasukkan ke pin ADC Arduino untuk pembacaan. Skematik dari rangkaian sensor arus ditunjukkan pada Gambar 3.5 di bawah ini.



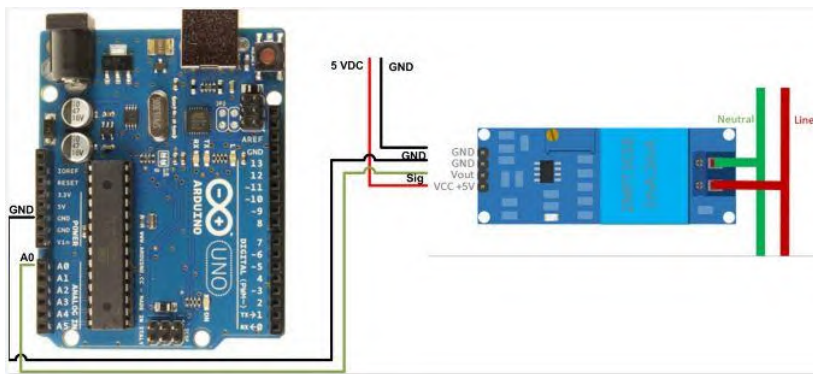
Gambar 3.5 Skematik rangkaian sensor arus LA 55-P

Berdasarkan datasheet sensor arus LA 55-P, resistor yang digunakan dalam rangkaian ini berkisar antara 10-100 Ω , maka dipilih resistor 100 Ω . Sehingga, tegangan output sensor = $\text{arus} \times 100 / 1000$.

Pin konektor X1-3 merupakan pin output dari rangkaian ini yaitu berupa tegangan, sedangkan pin X1-1, X1-2, dan X1-4 merupakan pin catu daya yang masing-masing -12V, GND, dan +12V yang didapatkan dari rangkaian *Power Supply*.

3.2.3 Rangkaian Sensor Tegangan AC ZMPT101B

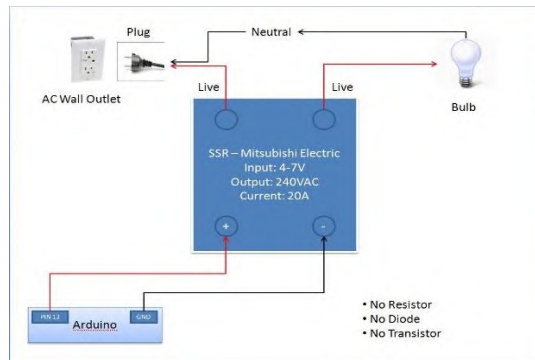
Pada pengukuran tegangan AC, digunakan sensor sejenis transformator kecil sebagai step down tegangan, yakni sensor ZMPT101B. Sensor ini membutuhkan suplai 5V. Output dari sensor tegangan ini berupa tegangan DC sinusoidal pada range 0 – 5V dengan nilai offset 2,5V, sehingga bisa langsung dimasukkan ke pin ADC Arduino untuk dilakukan pembacaan. Skematik rangkaian sensor tegangan AC ditunjukkan pada Gambar 3.6 di bawah ini



Gambar 3.6 Rangkaian sensor ZMPT101B

3.2.4 Rangkaian Solid State Relay

Sistem kontrol dalam tugas akhir ini menggunakan *Solid State Relay* (SSR) sebagai switch ON/OFF pada titik tertentu, yaitu pada input PLN, input Genset, dan saluran Torch. SSR ini dinilai paling cocok digunakan pada sistem ini karena mudah dioperasikan menggunakan mikrokontroler. Relay ini nantinya akan dioperasikan melalui web sehingga sangat erat hubungannya dengan pemrograman Arduino dan *Processing*. Berikut Gambar 3.7 merupakan rangkaian SSR yang akan diterapkan pada sistem.



Gambar 3.7 Rangkaian solid state relay

3.3 Perancangan Aplikasi Monitoring

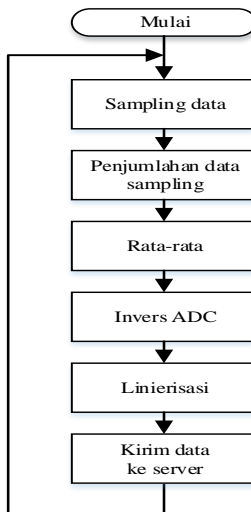
Perancangan Aplikasi Monitoring Charging Station meliputi perancangan metode pembacaan nilai arus dan tegangan DC pada arduino, perancangan metode pembacaan nilai arus dan tegangan AC pada Arduino, perancangan diagram use case, perancangan basis data, perancangan web service, dan perancangan antar muka sistem.

3.3.1 Perancangan Cloud Computing

Karena pada aplikasi sistem monitoring charging station berbasis cloud maka dibutuhkan *cloud computing* dengan jenis layanan *Platform as a Service* atau biasa disebut dengan *PaaS*. *PaaS* merupakan layanan dari *cloud computing* sebagai tempat atau lingkungan untuk menjalankan aplikasi sistem monitoring. Pada aplikasi sistem monitoring ini menggunakan *PaaS Heroku*. Karena dengan Heroku mendeploy aplikasi dapat dengan mudah di lingkungan *cloud* dengan dokumentasi konfigurasi yang cukup lengkap, *Heroku* juga kompatibel untuk aplikasi platform PHP sehingga aplikasi monitoring dapat terpasang pada *cloud Heroku*.

3.3.2 Perancangan Metode Pembacaan Nilai Arus dan Tegangan DC

Sebelum masuk ke Arduino, output dari rangkaian sensor yang dipasang akan dikondisikan menjadi tegangan dengan range 0 – 5V menggunakan rangkaian penurun arus dan tegangan agar bisa dimasukkan ke Arduino dengan aman. Pembacaan nilai arus dan tegangan DC dilakukan dengan mencari nilai rata-ratanya. Dilakukan beberapa sampling yang kemudian di rata-rata. Sampling dilakukan sebanyak 20 kali dengan delay 1 ms dan dijumlahkan. Hasil penjumlahan dibagi dengan 20 agar didapat nilai rata-rata. Hasil perhitungan rata-rata diambil untuk proses perhitungan selanjutnya. Metode pembacaan nilai arus dan tegangan dalam sistem ini ditunjukkan pada *flowchart* Gambar 3.8 di bawah ini

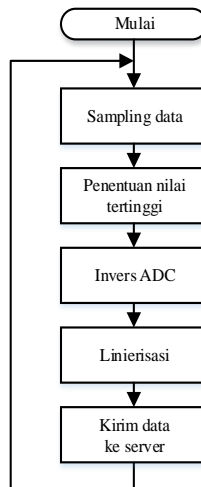


Gambar 3.8 Flowchart metode pembacaan nilai arus dan tegangan DC

3.3.3 Perancangan Metode Pembacaan Nilai Arus dan Tegangan AC

Untuk arus dan tegangan AC, tegangan input terlebih dahulu dipasangkan dengan *summing amplifier* agar tegangan yang terbaca pada pin Arduino bernilai positif, yaitu berada pada range 0 – 5V. Output dari *summing amplifier* masih berupa sinyal sinusoidal dengan tegangan *offset* 2,5 V, sehingga pembacaan arus dan tegangan AC ini menggunakan metode pencarian nilai tegangan puncaknya (V_p). Dilakukan sampling sebanyak 300 data dan nilai tertingginya akan diambil untuk proses selanjutnya.

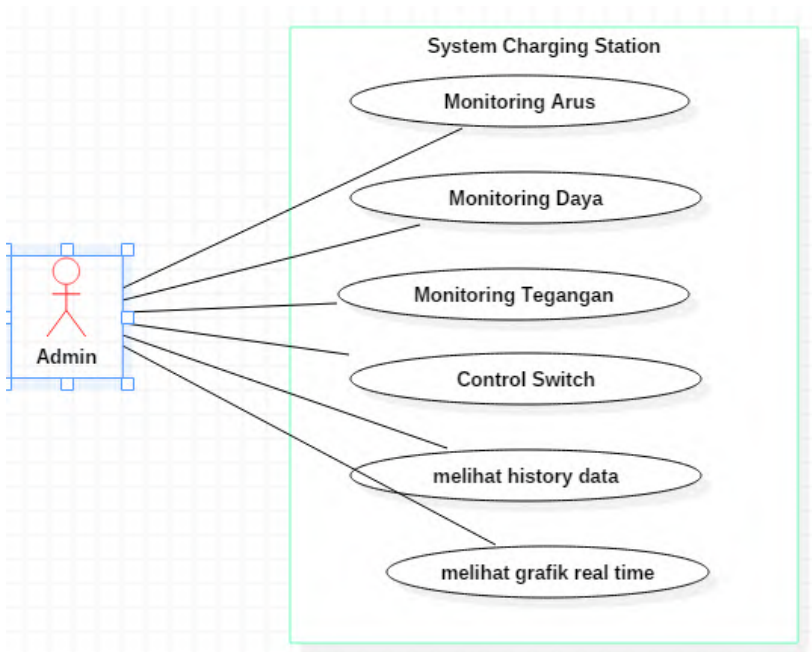
Arduino mengkonversi nilai pembacaan sensor menjadi nilai ADC dengan range 0 – 1023, sehingga digunakan rumus invers dari ADC untuk mengubah nilai ADC menjadi nilai tegangan sebenarnya. Setelah itu dilakukan linierisasi menggunakan regresi linier untuk mendapatkan nilai arus dan tegangan yang sebenarnya. Metode pembacaan arus dan tegangan AC ini ditunjukkan pada *flowchart* Gambar 3.9 di bawah ini.



Gambar 3.9 Flowchart metode pembacaan nilai arus dan tegangan AC

3.3.4 Perancangan Diagram Use Case

Diagram *use case* untuk aplikasi monitoring dapat dilihat pada gambar 3.10.



Gambar 3.10 use case diagram

Dalam aplikasi monitoring tersebut, admin dapat melakukan 6 hal inti, yang pertama adalah admin dapat melakukan monitoring arus listrik yang keluar dari charging station yang telah dipasang komponen jaringan sensor. Admin dapat melihat perubahan arus yang keluar secara real time. Kedua, admin dapat melakukan monitoring daya listrik yang keluar dari charging station. Ketiga, admin dapat melakukan monitoring tegangan listrik yang keluar dari charging

station. Keempat, admin dapat melakukan control switch ON atau OFF untuk melakukan pemutusan aliran arus listrik yang mengalir. Kelima, admin dapat melihat seluruh history data pada tabel database monitoring yang sudah disiapkan. Ke-enam admin dapat melihat grafik arus secara *real time* Seluruh sistem monitoring charging station ini disajikan secara real time sehingga memudahkan dalam melihat data keluaran yang didapat baik berupa arus tegangan ataupun daya.

3.3.5 Perancangan Basis Data

Pada perancangan basis data pada aplikasi sistem monitoring pada charging station ini hanya diperlukan dua tabel master yang dinamakan tabel *tb_charging* dan yang kedua adalah *tb_user* yang digunakan untuk menyimpan akun pengguna, sehingga tidak sembarang orang dapat mengakses aplikasi monitoring charging station yang dapat dilihat pada tabel 3.1 dan tabel 3.2

Tabel 3.1 Perancangan tb_user

No.	Nama Atribut	Tipe Data	Keterangan
1	<i>username</i>	<i>Varchar</i>	Sebagai <i>primary key</i> untuk akun
2	<i>password</i>	<i>varchar</i>	password yang disimpan merupakan <i>hash</i> dari password yang asli

Tabel 3.2 Perancangan tb_charging

No.	Nama Atribut	Tipe Data	Keterangan
1	<i>Id</i>	<i>integer</i>	<i>autoincrement</i>
2	<i>PV_arus</i>	<i>float</i>	Nilai arus yang mengalir pada panel surya
3	<i>PV_tegangan</i>	<i>float</i>	Nilai tegangan yang mengalir pada panel surya
4	<i>PV_daya</i>	<i>float</i>	Nilai daya yang mengalir pada panel surya

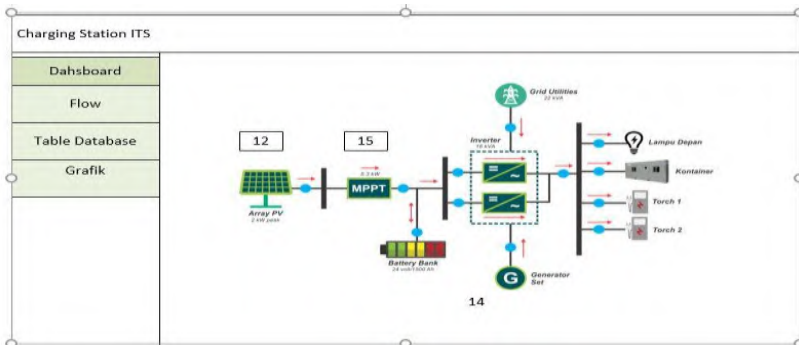
5	<i>Bat_arus</i>	<i>float</i>	Nilai arus yang mengalir pada baterai
7	<i>Bat_tegangan</i>	<i>float</i>	Nilai tegangan yang mengalir pada baterai
8	<i>Bat_daya</i>	<i>float</i>	Nilai daya yang mengalir pada baterai
9	<i>MPPT_arus</i>	<i>float</i>	Nilai arus yang mengalir pada MPPT
10	<i>MPPT_tegangan</i>	<i>float</i>	Nilai tegangan yang mengalir pada MPPT
4	<i>MPPT_daya</i>	<i>float</i>	Nilai daya yang mengalir pada MPPT
5	<i>Inverter_arus</i>	<i>float</i>	Nilai arus yang mengalir pada inverter
4	<i>Inverter_tegangan</i>	<i>float</i>	Nilai tegangan yang mengalir pada inverter
5	<i>Inverter_daya</i>	<i>float</i>	Nilai daya yang mengalir pada inverter
4	<i>Container_arus</i>	<i>float</i>	Nilai arus yang mengalir pada container
5	<i>Container_tegangan</i>	<i>float</i>	Nilai tegangan yang mengalir pada container
4	<i>Container_daya</i>	<i>float</i>	Nilai daya yang mengalir pada container
5	<i>Torch_1_arus</i>	<i>float</i>	Nilai arus yang mengalir pada torch 1
4	<i>Torch_1_tegangan</i>	<i>float</i>	Nilai tegangan yang mengalir pada torch 1
4	<i>Torch_1_daya</i>	<i>float</i>	Nilai daya yang mengalir pada torch 1
5	<i>Torch_2_arus</i>	<i>float</i>	Nilai arus yang mengalir pada torch 2
6	<i>Torch_2_tegangan</i>	<i>float</i>	Nilai tegangan yang mengalir pada torch 2
7	<i>Torch_2_daya</i>	<i>float</i>	Nilai daya yang mengalir pada torch 2
8	<i>waktu</i>	<i>timestamp</i>	Current time untuk mencatat waktu

3.3.6 Perancangan Web Service

Web Service pada aplikasi sistem monitoring ini digunakan untuk mengirim data dari komponen sensor *Charging Station* melalui mikrokontroller *Arduino* kepada aplikasi monitoring tersebut, pengirim ini menggunakan *Ethernet shield* yang terhubung ke jaringan internet karena data yang dikirimkan oleh sensor-sensor tersebut harus ditampilkan secara real time. Web Service yang dirancang adalah Web Service sederhana yang memanfaatkan metode *POST* pada *http*, sehingga komponen sensor hanya perlu memanggil url dari web service ini dan menambahkan parameter-parameter. Parameter-parameter yang digunakan pada program *Arduino* disamakan dengan field pada database misalkan *pv_arus* pada *Arduino* harus sama dengan yang terdapat pada Web Service tersebut menggunakan *insert into value POST*. Selanjutnya data tersebut akan disimpan didalam database pada tabel *tb_charging*.

3.3.7 Perancangan Antarmuka Sistem

Sesuai dengan perancangan pada diagram *use case*, admin dapat menjalankan 6 fitur utama, yaitu dapat melihat monitoring arus, monitoring daya, monitoring tegangan, control switch, dan melihat tabel database. Masing-masing fitur tersebut akan dirancang antarmuka sistem yang sederhana dan mudah dipahami oleh pengguna. Perancangan antarmuka sistem monitoring dan kontrol dalam Tugas Akhir ini menggunakan *interface*. Aplikasi monitoring ini terkoneksi dengan database server sehingga data yang ditampilkan pada dashboard antarmuka sistem merupakan data terakhir yang masuk ke database. Sedangkan data sebelumnya akan tetap tersimpan di *MySQL* database server. Tampilan antarmuka sistem *Charging Station* ITS ditunjukkan pada Gambar 3.11.



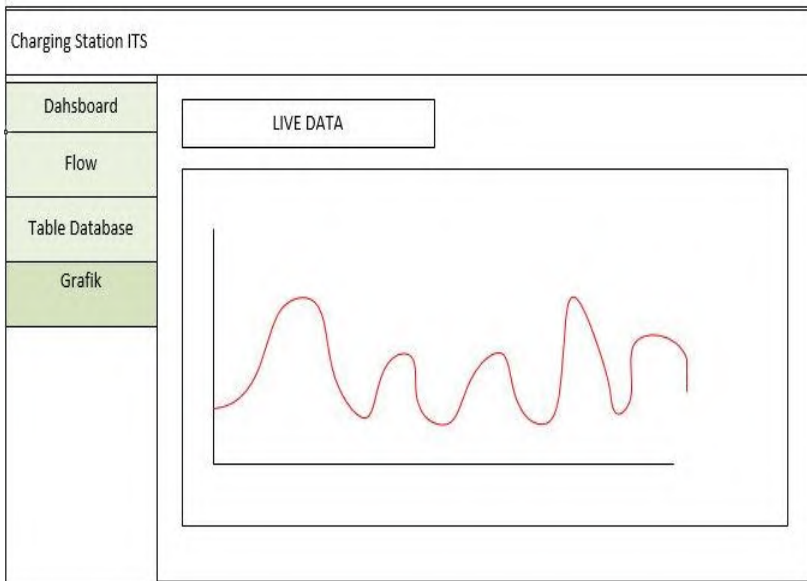
Gambar 3.11 Tampilan dashboard pada aplikasi monitoring *charging station ITS*

Sistem pengambilan data dilakukan secara *realtime*, sehingga data yang ditampilkan pada antarmuka sistem merupakan kondisi sistem terkini yang terdeteksi oleh sensor terpasang. Hasil pembacaan sensor dikirim tiap 10 detik sekali oleh mikrokontroler *Arduino*. Sistem monitoring ini juga bersifat *data logger*, karena pada tampilannya juga menyediakan tab yang berisi database hasil pembacaan sensor sebelumnya. Berikut Gambar 3.12 merupakan tampilan database pada aplikasi monitoring *Charging Station*.

Charging Station ITS			
Dahsboard	Table data monitoring		
Flow			
Table Database			
Grafik			
No	Arus PV	Arus Torch	

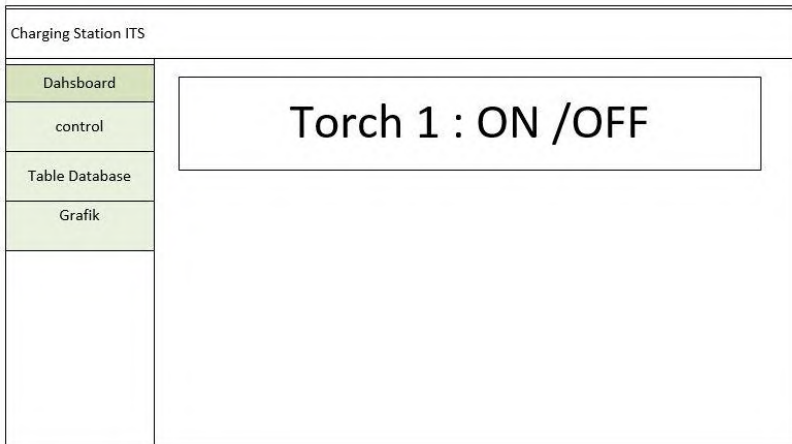
Gambar 3.12 Tampilan database pada aplikasi monitoring *charging station ITS*

Data hasil pembacaan sensor akan tersimpan dalam database sehingga *user* bisa melihat *history* data kapanpun. Untuk kemudahan dalam melihat karakteristik data dari sistem yang dimonitor, disediakan pula tampilan grafik data seperti pada Gambar 3.13 .



Gambar 3.13 Tampilan tab grafik pada aplikasi monitoring *charging station* ITS

Grafik pada aplikasi monitoring ini berjalan secara *realtime*. Selain itu, pada aplikasi monitoring sistem monitoring dan kontrol charging station terdapat juga sub tab kontrol *switch* ON/OFF untuk mengontrol relay terpasang, sehingga memudahkan user untuk melakukan monitoring maupun kontrol pada charging station dengan hanya membuka satu alamat web. Berikut Gambar 3.14 merupakan tampilan tab kontrol *charging station*.



Gambar 3.14 Tampilan tab kontrol pada aplikasi monitoring *charging station ITS*

Ketika tombol 'ON' ditekan, maka saluran pada *Torch* akan tersambung (*close*), dan saat tombol 'OFF' ditekan maka saluran *Torch* tidak tersambung atau dalam keadaan terputus (*open*)

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Pada implementasi sistem, digunakan beberapa perangkat pendukung sebagai berikut:

1. Arduino Mega 2560 dan *Ethernet Shield*
2. Laptop ASUS intel Pentium dual-core *processor* (2.10 Ghz) dengan 4 GB RAM
3. Arduino *Development Kit* versi 1.6.3 sebagai IDE untuk mengimplementasikan kode program arduino
4. Sublime sebagai *text editor* untuk mengimplementasikan kode program aplikasi monitoring
5. Kontrol switch ON/OFF menggunakan *Solid State Relay* (SSR)
6. Router yang digunakan untuk membuat jaringan lokal ialah TP-LINK.
7. Modem smartfren 3G.

4.2 Implementasi

Pada subbab ini akan dijelaskan implementasi yang telah dilakukan pada sistem. Terdapat dua lingkup implementasi yang akan dijelaskan pada subbab ini, yaitu implementasi mikrokontroler dan komponen sensor, dan implementasi aplikasi monitoring. Pada masing-masing lingkup tersebut juga akan dijelaskan mengenai fungsi-fungsi utama yang digunakan sehingga fungsionalitas sistem dapat bekerja dengan baik.

4.2.1 Implementasi Mikrokontroler dan Komponen Sensor

Pada mikrokontroler arduino, secara *default* program harus memiliki fungsi *setup* dan *loop*. Fungsi *setup* berguna untuk pengaturan mikrokontroler pada saat mikrokontroler tersebut melakukan *booting*, sedangkan fungsi *loop* berisi tentang perintah-perintah yang akan dieksekusi ketika mikrokontroler tersebut sedang *running*.

4.2.1.1 Implementasi Mikrokontroler metode pembacaan nilai arus dan tegangan AC

Pada implementasi Metode pembacaan arus dan tegangan AC ini menggunakan metode pencarian nilai tegangan puncaknya (V_p). Dilakukan sampling sebanyak 300 data dan nilai tertinggi akan diambil untuk proses selanjutnya. Berikut kode sumber 4.1 dibawah ini.

1	<code>for(i = 0; i < 300; i++) {</code>
2	<code> start_times[i] = micros();</code>
3	<code> values[i] = analogRead(A0);</code>
4	<code> if (values[i] >= peak0) {</code>
5	<code> peak0 = values[i];</code>
6	<code> }</code>
7	<code> stop_times[i] = micros();</code>
8	<code>}</code>
9	<code> adc0 = (peak0 * 5) / 1023 ;</code>
10	<code> linear0 = (adc0 - 2.5221) / 0.0103 ;</code>

Kode sumber 4.1 implementasi pembacaan nilai arus dan tegangan AC

4.2.1.2 Implementasi Mikrokontroler metode pembacaan nilai arus dan tegangan DC

Pada implementasi metode Pembacaan nilai arus dan tegangan DC dilakukan dengan mencari nilai rata-ratanya. Dilakukan beberapa sampling yang kemudian di rata-rata. Sampling dilakukan sebanyak 20 kali dengan delay 1 ms dan dijumlahkan. Hasil penjumlahan dibagi dengan 20 agar didapat nilai rata-rata. Hasil perhitungan rata-rata diambil untuk proses perhitungan selanjutnya. Berikut kode sumber 4.2 di bawah ini.

1	for (i = 0; i < sampling; i++) {
2	sum15 = sum15 + analogRead(A15);
3	delay(1);
4	}
5	average15 = sum15 / sampling;
6	adc15 = average15 * 5 / 1023;
7	linear15 = (adc15 - 0.1468) / 0.1602;
8	delay(1);
9	sum15 = 0;

Kode sumber 4.2 implementasi pembacaan nilai arus dan tegangan DC

4.2.1.3 Implementasi Web Service pada Mikrokontroler

Data-data sensor dari perhitungan di Arduino akan diteruskan ke cloud server dengan cara mengirimkannya melalui web service. Pada program di Arduino dengan menggunakan library dari Ethernet terdapat metode pengiriman data sensor tersebut menggunakan http POST, metode ini harus menyertakan juga alamat url yang dituju tetapi sebelum itu kita harus mengumpulkan semua data-data dari beberapa sensor tersebut pada satu variable pada aplikasi ini menggunakan variable txData2. Variable tersebut berisi data-data sensor yang disesuaikan dengan nama insert yang terdapat pada web service di PHP sehingga data-data sensor dapat masuk ke dalam database. Implementasi dari web service dapat dilihat pada kode sumber 4.3 di bawah ini.

```

1  EthernetClient client;
2  if (client.connect(server, 80)){
3      txData2 = "ac_inv_tegangan_1="+ (String
4      (ac_inv_tegangan_1));
5
6      Serial.println("connected");
7      Serial.print(txData2);
8      client.println("POST/insert2.php
9      HTTP/1.1");
10     client.println("Host:
11     www.chargingstation.its.ac.id");
12     client.println("Connection: close");
13     client.print("Content-Type: application/x-
14     www-form-urlencoded\n");
15     client.print("Content-Length: ");
16     client.print(txData2.length());
17     client.print("\n\n");
18     client.print(txData2);
19     Serial.println(txData2);
20     delay (1000)
21     }
22
23     else{
24     Serial.println("Connection Failed.");
25     Serial.println();
26     delay (1000);
27     }
28     }

```

Kode sumber 4.3 implementasi web service pada arduino

4.2.1.3 Implementasi Kontrol Relay pada mikrokontroler

Pada aplikasi sistem monitoring charging station ini juga terdapat fitur kontrol dengan menjalankan mikrokontroler processing. Kontrol ini memakai komponen solid state relay yang tersambung pada Arduino dan arus listrik yang mengalir pada sensor-sensor yang telah terpasang. Fitur kontrol ini berfungsi untuk admin yang ingin memutuskan dan menyambungkan arus yang mengalir. Pada aplikasi monitoring ditambahkan file.txt yang berguna untuk

mencatat, Jika kita menekan ON pada aplikasi monitoring ini program akan mengirimkan trigger ke file txt tersebut yang dilogikakan dengan perintah 1 dan jika OFF dengan perintah 0. Kemudian processing akan membacanya dan melakukan trigger kepada Arduino sehingga merubah relay menjadi ON ataupun OFF. Implementasi control Arduino dapat dilihat pada kode sumber 4.4 di bawah ini

```

1  import processing.serial.*;
2  Serial port;
3  void setup() {
4    port = new Serial(this, Serial.list()[0],
5    9600);
6  }
7  void draw() {
8    String onoroff[] =
9    loadstring("http://chargingstation.its.ac.i
10   d/LEDstate.txt");
11   print(onoroff[0]);
12
13   if (onoroff[0].equals("1") == true) {
14     println(" - TELLING ARDUINO TO TURN LED
15     ON");
16     port.write('H'); // Send "H" over
17     serial to set LED to HIGH
18   }
19
20   if (onoroff[0].equals("0") == true){
21
22     println(" - TELLING ARDUINO TO TURN LED
23     OFF");
24     port.write('L'); // Send "L" over
25     serial to set LED to LOW
26   }
27   delay(1000);
28
29 }

```

Kode sumber 4.4 implementasi kontrol pada Arduino

4.2.2 Implementasi Aplikasi Monitoring

Disini akan dijelaskan mengenai implementasi dari aplikasi monitoring charging station karena inti dari aplikasi ini adalah bagaimana melakukan monitoring terhadap charging station secara real time dapat dipastikan pengiriman datanya juga terjadi secara real time juga.

4.2.2.1 Implementasi Cloud Computing Heroku

Sebelum melakukan deploy aplikasi menggunakan Heroku terlebih dahulu membuat akun pada heroku selanjutnya pastikan PHP, composer dan git sudah terinstall pada komputer. Berikut kode sumber untuk dijalankan pada *command prompt* atau *git bash*.

1	\$ heroku login
2	
3	\$ git clone https://github.com/heroku/php-
4	getting-started.git
5	
6	\$ cd php-getting-started
7	
8	\$ heroku create
9	\$ git push heroku master
10	
11	\$ heroku open

Kode sumber 4.5 implementasi Heroku

4.2.2.2 Implementasi Pengiriman Data ke Database

Setelah mikrokontroler melakukan pengiriman data, data tersebut harus diterima oleh web service. Selanjutnya data sensor diterima dengan menyiapkan query untuk meneruskannya ke database. Memakai perintah insert dan menggunakan value post. Implementasi pada pengiriman data ke database dapat dilihat pada kode sumber 4.5 di bawah ini.

1	require("config3.php");
2	\$query = "INSERT INTO tbcharging_2 (
3	pv_arus,
4	pv_tegangan,
5	pv_daya,
6	mppt_arus,
7	mppt_tegangan,
8	mppt_daya,
9	bat_arus)
10	VALUES
11	(
12	'\$_POST[pv_arus]',
13	'\$_POST[pv_tegangan]',
14	'\$_POST[pv_daya]',
15	'\$_POST[mppt_arus]',
16	'\$_POST[mppt_tegangan]',
17	'\$_POST[mppt_daya]',
18	'\$_POST[bat_arus]'
19);

Kode sumber 4.6 implementasi pengiriman data ke database

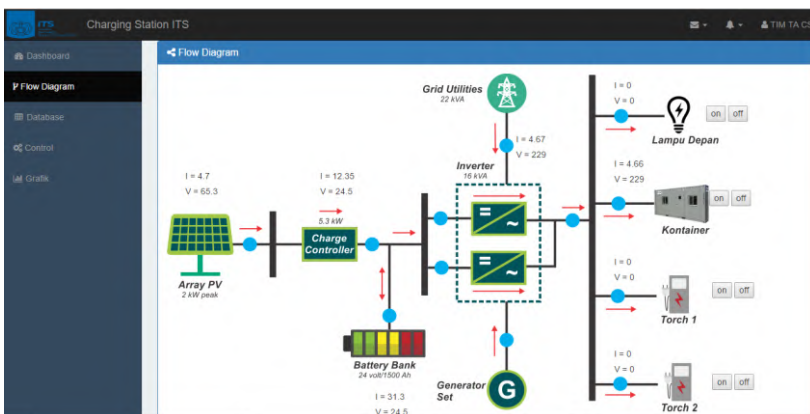
4.2.2.3 Implementasi Fitur Monitoring

Pada fitur monitoring terdapat fungsi untuk menampilkan data secara real time agar jika terjadi perubahan sewaktu-waktu aplikasi ini dapat mengikuti perubahan itu tanpa harus mereload page. Sebelum menampilkannya, Pertama-tama aplikasi mengambil data sensor yang paling terakhir dimasukkan oleh Arduino dari database selanjutnya data ditampilkan menggunakan javasript yang mereload setiap 3 detik dengan menggunakan fungsi *setInterval()*. Berikut implementasi fitur monitoring yang terdapat pada kode sumber 4.6 dan halaman fitur monitoring yang terdapat pada gambar 4.1 dibawah ini.

1	function auto_load()
2	\$.ajax({
3	url: "torch2.php",
4	
5	cache: false,

6	success: function(data){
7	\$("#torch2con").html(data);
8	
9	}
10	});
11	}
12	
13	\$(document).ready(function(){
14	
15	auto_load();
16	});
17	
18	setInterval(auto_load,3000);

Kode sumber 4.7 implementasi monitoring real time



Gambar 4.1 halaman sistem monitoring charging station

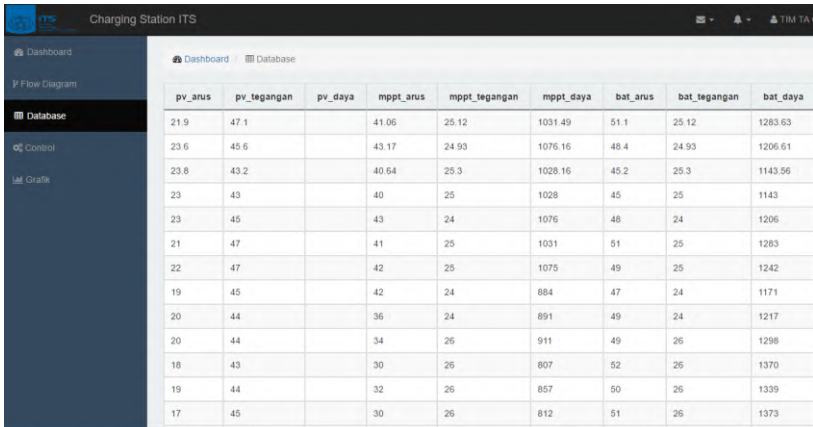
4.2.2.4 Implementasi Fitur Melihat History Data

Pada aplikasi monitoring ini juga dapat melihat history data-data sensor yang sebelumnya pada tabel yang telah disiapkan. Sebelum menampilkannya, aplikasi mengambil semua data dari database selanjutnya hasil pengambilan dari database tersebut kita rubah ke dalam bentuk JSON agar mudah untuk menampilkannya kedalam format tabel. Tabel disini menggunakan datatables javascript

dan juga dengan menambahkan perubahan secara real time, implementasinya dapat dilihat pada kode sumber 4.7 dan juga gambar 4.2 dibawah ini.

1	var table = \$('#test').DataTable({
2	
3	"retrieve": true,
4	"searching": false,
5	"paging": false,
6	"ordering": false,
7	"ajax": "config2.php",
8	"columns": [
9	{ "data": "arus1" },
10	{ "data": "tegangan1" },
11	{ "data": "daya1" },
12	{ "data": "arus2" },
13	{ "data": "tegangan2" },
14	{ "data": "daya2" },
15]
16	});
17	
18	
19	
20	setInterval(function () {
21	table.ajax.reload(null, false);
22	}, 3000);
23	});

Kode sumber 4.8 implementasi melihat history data



pv_arus	pv_tegangan	pv_waktu	mppt_arus	mppt_tegangan	mppt_waktu	bat_arus	bat_tegangan	bat_waktu
21.9	47.1		41.06	25.12	1031.49	51.1	25.12	1283.63
23.6	45.6		43.17	24.93	1076.16	48.4	24.93	1206.61
23.8	43.2		40.64	25.3	1028.16	45.2	25.3	1143.56
23	43		40	25	1028	45	25	1143
23	45		43	24	1076	48	24	1206
21	47		41	25	1031	51	25	1283
22	47		42	25	1075	49	25	1242
19	45		42	24	884	47	24	1171
20	44		36	24	891	49	24	1217
20	44		34	26	911	49	26	1298
18	43		30	26	807	52	26	1370
19	44		32	26	857	50	26	1339
17	45		30	26	812	51	26	1373

Gambar 4.2 halaman history data charging station

4.2.2.5 Implementasi Melihat Grafik Real Time

Agar dapat memudahkan admin dalam melakukan monitoring charging station secara langsung maka dibuat fitur untuk melihat grafik secara real time, grafik pada aplikasi monitoring mencatat data perubahan dan membuat grafik berdasarkan waktu dan arus tegangan yang keluar pada beban setiap 1 detik, grafik ini menggunakan Highchart.js berupa satu garis line. Pertama-tama aplikasi mengambil data dari database yang harus diambil hanya arus dan waktu berupa time stamp selanjutnya data tersebut dijadikan ke dalam bentuk JSON, bentuk JSON harus mengikuti format yang seharusnya untuk menerapkan pada highchart.js setelah itu data ditampilkan dengan menggunakan fungsi *setTimeout()*. Bias dilihat pada kode sumber 4.9.,4.10 dan gambar 4.3 dibawah ini

```

1 while ($row = mysql_fetch_array($fetch,
2     MYSQL_ASSOC))
3 {
4     $row_array['waktu'] = $row['waktu'];
5     $row_array['arus1'] = $row['arus1'];
6     array_push($return_arr,$row_array);
7 }

```

```

8      $data=json_encode($return_arr);
9
10     $mapper = function($item) {
11         Return          array(strtotime($item-
12 >waktu)*1000, $item->arus1);
13     };
14
15     $json_var=    json_encode(array_map($mapper,
16 json_decode($data)));
17     $input = $json_var;
18     $input = str_replace( '"', "'", $input );
19     $input = str_replace( '[', "'", $input );
20     $input = str_replace( ']', "'", $input );
21     echo '[';
22     echo $input;
23     echo ']';

```

Kode sumber 4.9 implementasi grafik JSON

```

1      function requestData() {
2          $.ajax({
3              url: 'grafikjson.php',
4              success: function(point) {
5                  var series = chart.series[0],
6                      shift = series.data.length > 20;
7                  chart.series[0].addPoint(point,          true,
8                  shift);
9                      setTimeout(requestData, 1000);
10              },
11              cache: false
12          });
13      }

```

Kode sumber 4.10 implementasi grafik pada aplikasi monitoring



Gambar 4.3 halaman grafik grafik real time

BAB V

UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba yang meliputi uji tingkat akurasi dan uji kinerja serta analisa setiap pengujian.

5.1 Lingkungan Uji Coba

Pada subbab ini menjelaskan tentang lingkungan yang digunakan untuk menguji fungsionalitas dan tingkat akurasi dari implementasi sistem yang telah dirancang. Lingkungan uji coba menggunakan bantuan perangkat lunak dan perangkat keras dengan spesifikasi sebagai berikut:

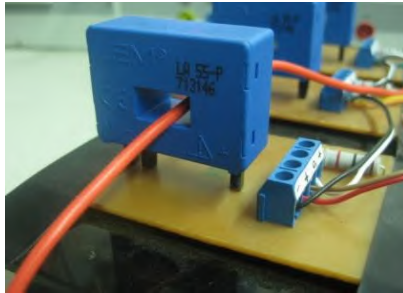
1. Perangkat Keras
 - a. Laptop dengan prosessor Intel Core i3 CPU @
 - b. RAM : 4,00 GB
 - c. Tipe sistem : 64-bit sistem operasi
 - d. Clamp Ampere
 - e. Router TP Link
 - f. Modem Smartfren 3G
2. Perangkat Lunak
 - a. Sistem Operasi : *Windows 8.1*
 - b. *IDE : arduino 1.6.3*
 - c. Sebuah *web browser*

5.2 Skenario Uji Coba

Uji coba ini dilakukan untuk menguji apakah fungsionalitas program telah diimplementasikan dengan benar dan berjalan sesuai rencana, dan juga tingkat akurasi dan error.

5.2.1 Pengujian Linearisasi Sensor LA 55-P

Sensor arus LA 55-P digunakan untuk mengukur arus AC yang mengalir pada konduktor yang dilewatkan pada lubang di tengahnya. Rangkaian sensor arus LA 55-P ditunjukkan pada Gambar 4.1 di bawah ini.



Gambar 5.1 Rangkaian sensor arus LA 55-P

Untuk mengetahui karakteristik sensor arus LA 55-P, dilakukan pengujian menggunakan *AC Power Supply* di laboratorium dengan *load bank* berupa lampu dengan total sebesar 1500 Watt seperti yang terlihat pada Gambar 4.2 di bawah ini.



Gambar 5.2 Lampu 1500 Watt untuk pengujian

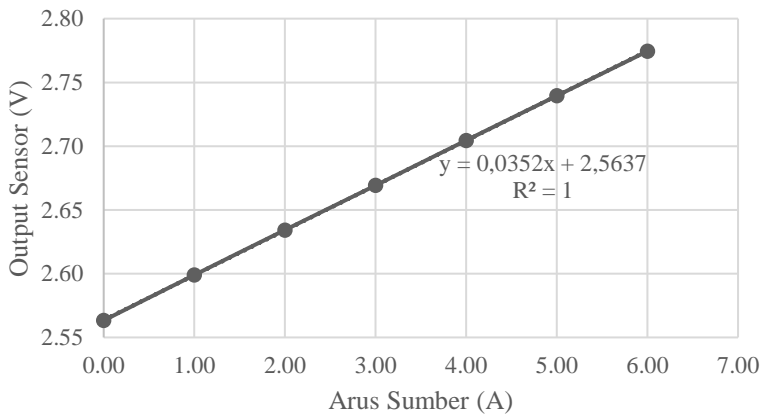
Pengujian dilakukan pada masing-masing sensor LA 55-P yang akan dipasang pada *Charging Station*, lebih tepatnya pada saluran lampu depan, inverter, kontainer, *Torch 1*, dan *Torch 2*.

5.2.1.1 Linierisasi LA 55-P untuk Lampu Depan

Data hasil pengujian sensor arus LA 55-P yang akan dipasang pada lampu depan *Charging Station* ITS dipaparkan pada Tabel 5.1, sedangkan grafik hasil pengujian ditunjukkan pada Gambar 5.3 di bawah ini.

Tabel 5.1 Hasil pengujian rangkaian sensor arus LA 55-P (lampu depan)

Arus Sumber (A)	Output Sensor (V)
0	2,56
1	2,60
2	2,63
3	2,67
4	2,70
5	2,74
6	2,77



Gambar 5.3 Grafik hasil pengujian sensor arus LA 55-P (lampu depan)

Dari data pengujian tersebut dilakukan linierisasi menggunakan regresi linier untuk mendapatkan persamaan tegangan output sensor terhadap arus sumber. Persamaan inilah yang akan dimasukkan ke mikrokontroler Arduino untuk mendapatkan nilai arus yang mendekati nilai sebenarnya.

Dari proses linierisasi dengan persamaan regresi linier, didapatkan koefisien 'm' adalah 0,0352 dan nilai koefisien 'c' adalah 2,5637.

Berdasarkan persamaan linier secara umum:

$$y = mx + c$$

dengan 'y' merupakan tegangan output sensor dan 'x' merupakan nilai arus sumber yang terukur, maka persamaan yang dimasukkan ke Arduino adalah sebagai berikut:

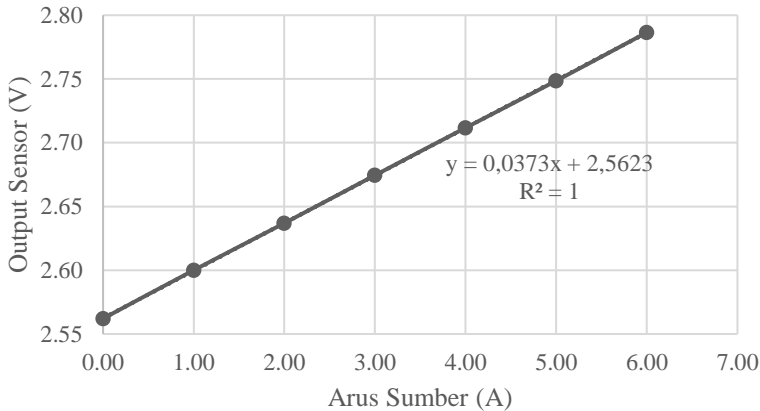
$$\text{arus terukur} = \frac{y - 2,5637}{0,0352}$$

5.2.1.2 Linearisasi LA 55-P untuk Inverter

Data hasil pengujian sensor arus LA 55-P yang akan dipasang pada inverter *Charging Station* ITS dipaparkan pada Tabel 5.2, sedangkan grafik hasil pengujian ditunjukkan pada Gambar 5.4 di bawah ini.

Tabel 5.2 Hasil pengujian rangkaian sensor arus LA 55-P (inverter)

Arus Sumber (A)	Output Sensor (V)
0	2,56
1	2,60
2	2,64
3	2,67
4	2,71
5	2,75
6	2,79



Gambar 5.4 Grafik hasil pengujian sensor arus LA 55-P (inverter)

Dari data pengujian tersebut dilakukan linierisasi menggunakan regresi linier, didapatkan koefisien ‘m’ adalah 0,0373 dan nilai koefisien ‘c’ adalah 2,5623.

Berdasarkan persamaan linier secara umum:

$$y = mx + c$$

dengan ‘y’ merupakan tegangan output sensor dan ‘x’ merupakan nilai arus sumber yang terukur, maka persamaan yang dimasukkan ke Arduino adalah sebagai berikut:

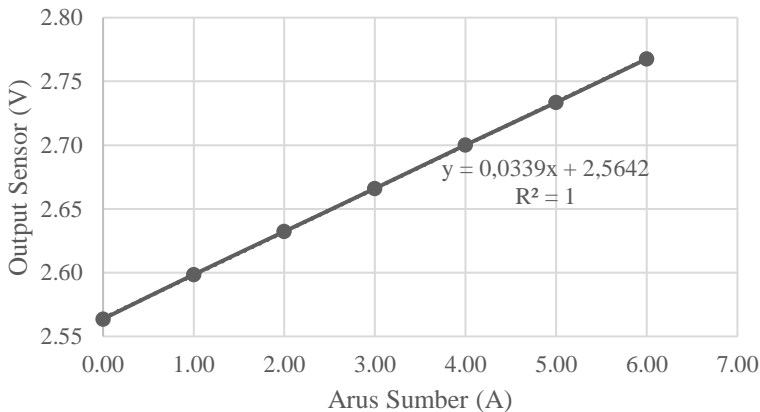
$$\text{arus terukur} = \frac{y - 2,5623}{0,0373}$$

5.2.1.3 Linearisasi LA 55-P untuk Kontainer

Data hasil pengujian sensor arus LA 55-P yang akan dipasang pada kontainer *Charging Station* ITS dipaparkan pada Tabel 5.3, sedangkan grafik hasil pengujian ditunjukkan pada Gambar 5.5 di bawah ini.

Tabel 5.3 Hasil pengujian rangkaian sensor arus LA 55-P (kontainer)

Arus Sumber (A)	Output Sensor (V)
0	2,56
1	2,60
2	2,63
3	2,67
4	2,70
5	2,73
6	2,77

**Gambar 5.5 Grafik hasil pengujian sensor arus LA 55-P (kontainer)**

Dari data pengujian tersebut dilakukan linierisasi menggunakan regresi linier, didapatkan koefisien ‘m’ adalah 0,0339 dan nilai koefisien ‘c’ adalah 2,5642.

Berdasarkan persamaan linier secara umum:

$$y = mx + c$$

dengan ‘y’ merupakan tegangan output sensor dan ‘x’ merupakan nilai arus sumber yang terukur, maka persamaan yang dimasukkan ke Arduino adalah sebagai berikut:

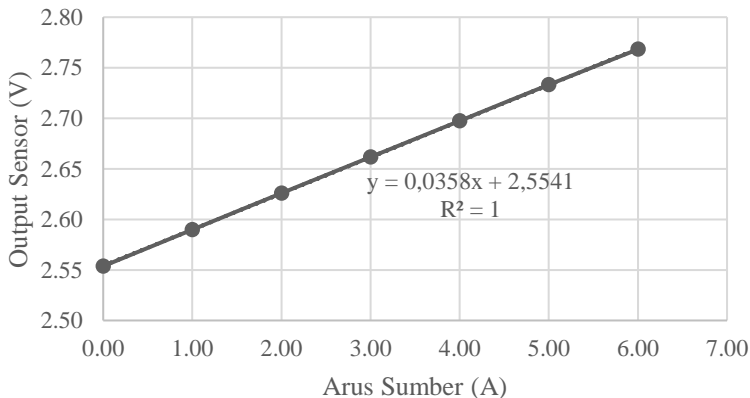
$$\text{arus terukur} = \frac{y - 2,5642}{0,0339}$$

5.2.1.4 Linearisasi LA 55-P untuk Torch 1

Data hasil pengujian sensor arus LA 55-P yang akan dipasang pada Torch 1 Charging Station ITS dipaparkan pada Tabel 5.4, sedangkan grafik hasil pengujian ditunjukkan pada Gambar 5.6 di bawah ini.

Tabel 5.4 Hasil pengujian rangkaian sensor arus LA 55-P (Torch 1)

Arus Sumber (A)	Output Sensor (V)
0	2,55
1	2,59
2	2,63
3	2,66
4	2,70
5	2,73
6	2,77



Gambar 5.6 Grafik hasil pengujian sensor arus LA 55-P (Torch 1)

Dari data pengujian tersebut dilakukan linierisasi menggunakan regresi linier, didapatkan koefisien 'm' adalah 0,0358 dan nilai koefisien 'c' adalah 2,5541.

Berdasarkan persamaan linier secara umum:

$$y = mx + c$$

dengan ‘y’ merupakan tegangan output sensor dan ‘x’ merupakan nilai arus sumber yang terukur, maka persamaan yang dimasukkan ke Arduino adalah sebagai berikut:

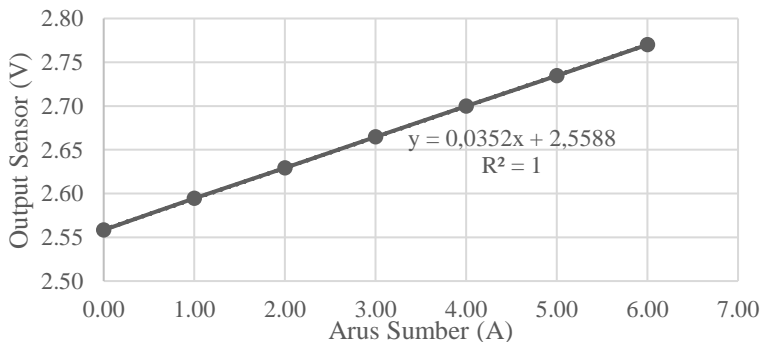
$$\text{arus terukur} = \frac{y - 2,5541}{0,0358}$$

5.2.1.5 Linearisasi LA 55-P untuk Torch 2

Data hasil pengujian sensor arus LA 55-P yang akan dipasang pada *Torch 2 Charging Station ITS* dipaparkan pada Tabel 5.5, sedangkan grafik hasil pengujian ditunjukkan pada Gambar 5.7 di bawah ini.

Tabel 5.5 Hasil pengujian rangkaian sensor arus LA 55-P (*Torch 2*)

Arus Sumber (A)	Output Sensor (V)
0	2,56
1	2,59
2	2,63
3	2,66
4	2,70
5	2,73
6	2,77



Gambar 5.7 Grafik hasil pengujian sensor arus LA 55-P (*Torch 2*)

Dari data pengujian tersebut dilakukan linierisasi menggunakan regresi linier, didapatkan koefisien 'm' adalah 0,0352 dan nilai koefisien 'c' adalah 2,5588.

Berdasarkan persamaan linier secara umum:

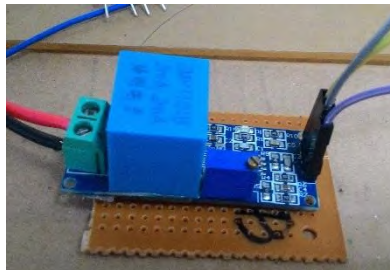
$$y = mx + c$$

dengan 'y' merupakan tegangan output sensor dan 'x' merupakan nilai arus sumber yang terukur, maka persamaan yang dimasukkan ke Arduino adalah sebagai berikut:

$$\text{arus terukur} = \frac{y - 2,5588}{0,0352}$$

5.2.1.6 Linearisasi Sensor Tegangan ZMPT101B

Sensor tegangan ZMPT101B digunakan untuk mengukur tegangan AC. Pada sensor ini sudah terdapat rangkaian summing amplifier yang menaikkan tegangan offset 2,5V. Bentuk rangkaian sensor tegangan ZMPT101B ditunjukkan pada Gambar 5.8 di bawah ini.



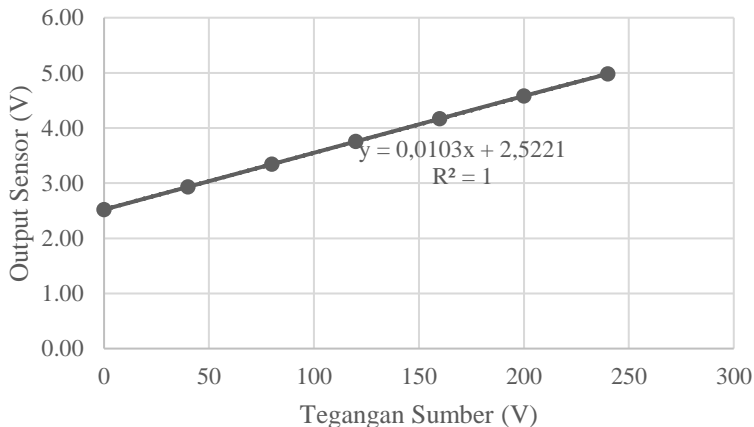
Gambar 4.11 Rangkaian sensor tegangan ZMPT101B

Untuk mengetahui karakteristik sensor tegangan ZMPT101B, dilakukan pengujian menggunakan *AC Power Supply* di laboratorium yang dihubungkan paralel dengan sensor.

Data hasil pengujian dipaparkan pada Tabel 5.6, sedangkan grafik hasil pengujian ditunjukkan pada Gambar 5.9 di bawah ini.

Tabel 5.6 Hasil pengujian rangkaian sensor tegangan ZMPT101B

Tegangan Sumber (V)	Output Sensor (V)
0	2,52
40	2,93
80	3,34
120	3,76
160	4,17
200	4,58
240	4,98

**Gambar 5.9** Grafik hasil pengujian sensor tegangan ZMPT101B

Dari data pengujian tersebut dilakukan linierisasi menggunakan regresi linier, didapatkan koefisien ‘m’ adalah 0,0103 dan nilai koefisien ‘c’ adalah 2,5221

Berdasarkan persamaan linier secara umum:

$$y = mx + c$$

dengan ‘y’ merupakan tegangan output sensor dan ‘x’ merupakan nilai tegangan sumber yang terukur, maka persamaan yang dimasukkan ke Arduino adalah sebagai berikut:

$$\text{tegangan terukur} = \frac{y - 2,5221}{0,0103}$$

5.2.1.7 Linearisasi Sensor Tegangan *Volatage Divider*

Pada tugas akhir ini, digunakan rangkaian pembagi tegangan atau Voltage Divider untuk mengukur tegangan DC. Dipilih R_2 sebesar 10 k Ω dan R_1 sebesar 47 k Ω untuk *Voltage Divider* yang akan dipasang pada bus DC untuk pengukuran tegangan baterai, inverter, dan *charge controller*. Sedangkan untuk mengukur tegangan PV, *Voltage Divider* dipasang pada bus PV dengan menggunakan R_2 sebesar 10 k Ω dan R_1 sebesar 200 k Ω . Nilai resistor tersebut disesuaikan dengan perhitungan agar tegangan outputnya berada pada range 0 – 5V sehingga aman untuk dimasukkan pada pin ADC arduino.

Untuk mengetahui karakteristik *Voltage Divider*, dilakukan pengujian menggunakan *DC Power Supply* di laboratorium yang dihubungkan paralel dengan sensor. Pengujian dilakukan pada masing-masing *Voltage Divider* yang akan dipasang pada *Charging Station ITS*, lebih tepatnya pada bus PV dan bus DC. Rangkaian *Voltage Divider* yang akan digunakan ditunjukkan pada Gambar 5.10 berikut.



Gambar 5.10 Rangkaian *Voltage Divider*

5.2.2 Pengujian pembacaan Sensor Arus LA 55-P pada *Charging Station ITS*

Sensor arus dilakukan pengujian untuk membandingkan nilai output sensor terhadap nilai input yang sebenarnya. Pengujian dilakukan langsung pada *Charging Station ITS*, tepatnya pada saluran inverter dan kontainer. Kabel dari sumber menuju beban dilewatkan pada sensor LA 55-P ini, maka arus yang mengalir akan terbaca pada sensor. Pemasangan sensor arus LA 55-P seperti ditunjukkan pada Gambar 5.11 di bawah ini.



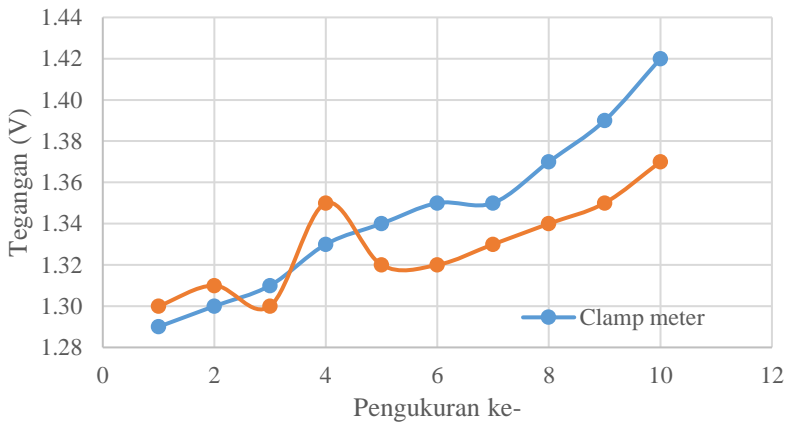
Gambar 5.11 Pemasangan LA 55-P pada kabel *Charging Station*

Data hasil pembacaan program setelah linierisasi yang ditampilkan pada *interface web* dibandingkan dengan pengukuran menggunakan *clamp meter*. Hasil pembacaan arus yang mengalir dari inverter terlihat pada Tabel 5.7 dan grafiknya pada Gambar 5.12 berikut ini.

Tabel 5.7 Hasil pengujian nilai arus terukur pada inverter

Pengukuran clamp meter (Ampere)	Pembacaan program (Ampere)	error (%)
1.29	1.30	0.78
1.30	1.31	0.77
1.31	1.30	0.76
1.33	1.35	1.50

1.34	1.32	1.49
1.35	1.32	2.22
1.35	1.33	1.48
1.37	1.34	2.19
1.39	1.35	2.88
1.42	1.37	3.52



Gambar 5.12 Grafik hasil pengujian nilai arus terukur pada inverter

5.2.2 Pengujian Pembacaan Sensor Tegangan ZMPT10B pada bus AC Charging Station ITS

Sensor tegangan dilakukan pengujian untuk membandingkan nilai output sensor terhadap nilai input yang sebenarnya. Pengujian dilakukan langsung pada *Charging Station ITS*, tepatnya pada bus AC. Tegangan pada bus AC akan terbaca oleh sensor saat sensor tegangan dipasang paralel dengan bus AC seperti pada Gambar 5.3 di bawah ini.

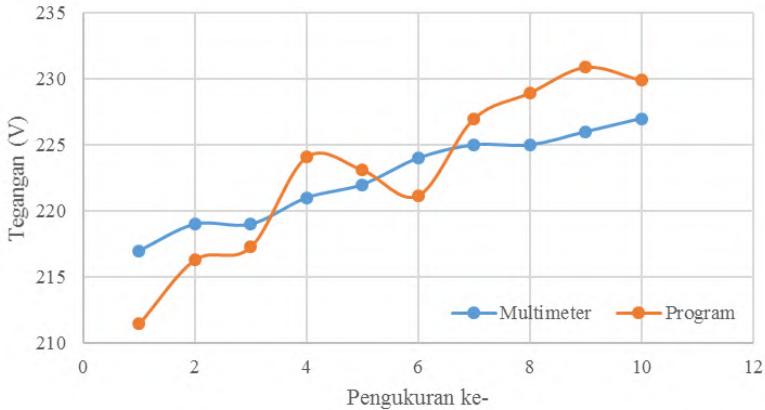


Gambar 5.13 Pemasangan ZMPT101B pada bus AC *Charging Station*

Data hasil pembacaan program setelah linierisasi yang ditampilkan pada *interface* web dibandingkan dengan pengukuran menggunakan multi meter. Hasil pembacaan tegangan terlihat pada Tabel 5.8 sedangkan grafiknya terlihat pada Gambar 5.3 berikut.

Tabel 5.8 Hasil pengujian nilai tegangan terukur pada bus AC

Pengukuran Multimeter (Ampere)	Pembacaan Program (Ampere)	Error (%)
217	211,45	2,56
219	216,30	1,23
219	217,27	0,79
221	224,07	1,39
222	223,10	0,49
224	221,16	1,27
225	226,98	0,88
225	228,92	1,74
226	230,86	2,15
227	229,89	1,27

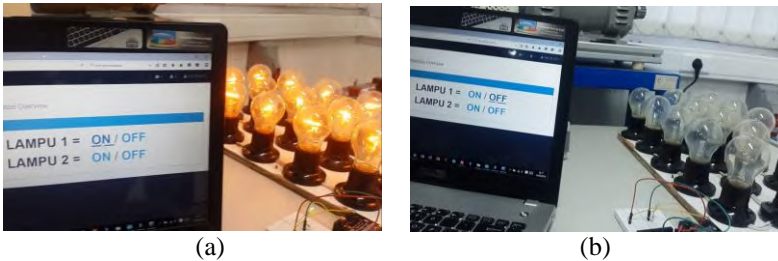


Gambar 5.14 Grafik hasil pengujian nilai tegangan terukur pada bus AC

Dari hasil di atas, terlihat bahwa error antara pembacaan sensor setelah linierisasi dengan tegangan sebenarnya cenderung kecil, yaitu 2,56% untuk error tertinggi. Hal ini bisa disebabkan tingkat presisi sensor yang digunakan kurang akurat ataupun proses linierisasi yang kurang tepat.

5.2.3 Pengujian Kontrol Relay SSR

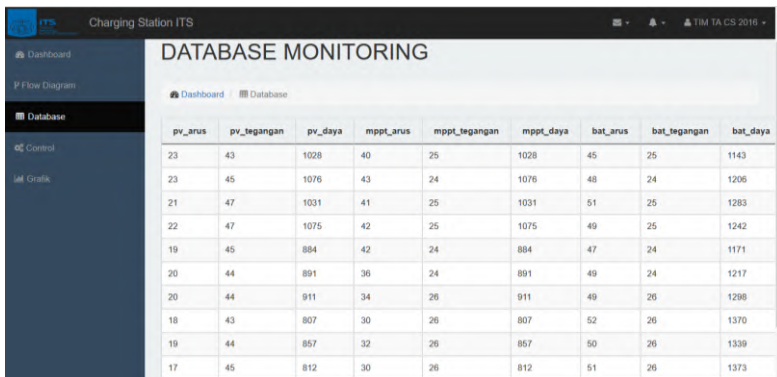
Pada tugas akhir ini terdapat kontrol relay sebagai switch ON/OFF pada titik tertentu, yaitu saluran menuju *Torch 1*, *Torch 2*, saluran dari PLN dan Genset. Relay yang digunakan adalah relay SSR yang bisa dikontrol menggunakan tegangan input DC antara 3 - 32V untuk mengontrol tegangan AC sampai 240V, sehingga tegangan output dari pin digital Arduino bisa langsung menjadi tegangan kontrolnya. Untuk mengetahui performa kontrol relay via web, dilakukan pengujian kinerja sistem kontrol pada beban lampu dan AC *Power Supply* laboratorium. Hasil pengujian kontrol relay via web ditunjukkan pada Gambar 5.15 di bawah ini.



Gambar 5.15 Pengujian kontrol relay via web saat (a) ON dan (b) OFF

5.2.4 Pengujian Data Antarmuka Aplikasi Monitoring

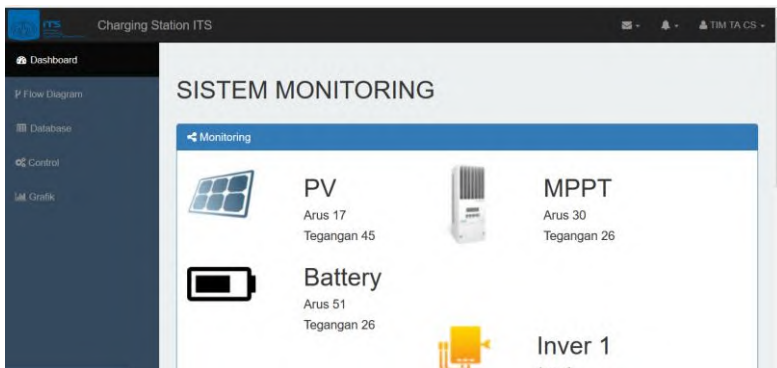
Pengujian ini dilakukan untuk melihat keberhasilan pengiriman data hasil pembacaan sensor arus dan tegangan dari Arduino ke server. Data terlebih dahulu tersimpan di database dan selanjutnya ditampilkan pada antarmuka aplikasi. Database menyimpan semua data yang dikirim oleh Arduino, sedangkan data dashboard pada antarmuka aplikasi *Charging Station* ini hanya menampilkan nilai terakhir yang masuk di database secara *realtime*. Berikut Gambar 5.16 menunjukkan tampilan data hasil pembacaan sensor yang telah tersimpan oleh database, Gambar 5.17 menunjukkan tampilan data dashboard web yang menampilkan nilai terakhir yang masuk di database, dan Gambar 5.18 menunjukkan tampilan *flow diagram* pada web *Charging Station ITS*.



pv_arus	pv_tegangan	pv_waktu	mppt_arus	mppt_tegangan	mppt_waktu	bat_arus	bat_tegangan	bat_waktu
23	43	1028	40	25	1028	45	25	1143
23	45	1076	43	24	1076	48	24	1206
21	47	1031	41	25	1031	51	25	1283
22	47	1075	42	25	1075	49	25	1242
19	45	884	42	24	884	47	24	1171
20	44	891	36	24	891	49	24	1217
20	44	911	34	26	911	49	26	1296
18	43	807	30	26	807	52	26	1370
19	44	857	32	26	857	50	26	1339
17	45	812	30	26	812	51	26	1373

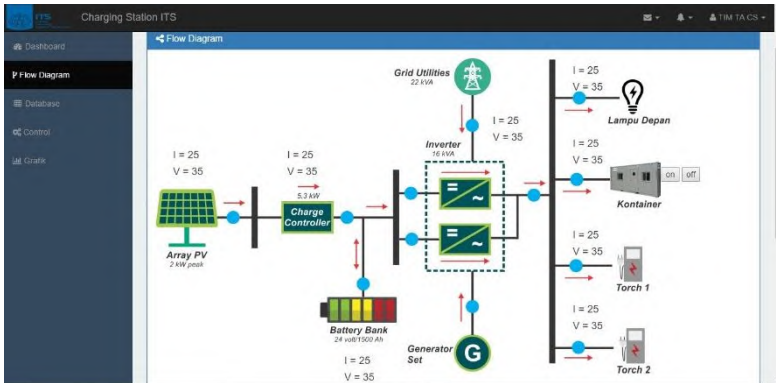
Gambar 5.16 Tampilan history data *Charging Station ITS*

Pada tampilan history data menampilkan data-data keluaran dari sensor arus tegangan berdasarkan waktu. Semua dapat dari database dapat dilihat pada menu perintah database.



Gambar 5.17 Tampilan data dashboard web *Charging Station ITS*

Pada pengujian tampilan dashboard data arus dan tegangan keluar secara periodik waktu. Sesuai dengan yang diharapkan terdapat beberapa komponen yang diukur diantaranya *PV*, *MMPT*, *Battery*, *Inverter*, *Kontainer*, *Torch 1*, dan *Torch 2*



Gambar 5.18 Tampilan *flow diagram* web *Charging Station ITS*

Pada pengujian tampilan monitoring flow diagram arus dan tegangan juga berubah secara real time dan terdapat tombol control yang dapat memutuskan arus yang mengalir melalui web.

BAB VI

KESIMPULAN DAN SARAN

1.1 Kesimpulan

Dari hasil pengamatan dan percobaan selama perancangan, implementasi, dan uji coba aplikasi, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem Monitoring *Charging Station* menggunakan *Arduino Mega 2560* dengan *Ethernet Shield* sebagai modul jaringan internet yang tersambung dengan sensor-sensor yang ditampilkan pada *Heroku Cloud Platform*.
2. Pada pembacaan sensor arus dan tegangan AC dilakukan teknik sampling dengan mengambil arus dan tegangan tertinggi dan untuk arus dan tegangan DC dengan cara menghitung arus rata-rata pada sampling tersebut dengan error rata-rata di bawah 5 %.
3. Sistem Monitoring *Charging Station* Menggunakan *Solid State Relay* yang disambungkan pada *Arduino* berfungsi untuk melakukan kontrol dengan cara menekan tombol ON/OFF sebagai trigger *Arduino Processing* untuk memutuskan arus dan tegangan pada obyek yang akan diukur.

1.2 Saran

Berdasarkan perancangan dan pengujian sistem oleh penulis, maka untuk pengembangan selanjutnya dari sistem monitoring dan kontrol ini adalah melengkapi fitur-fitur yang ditampilkan pada web, seperti irradiant, suhu baterai, SOC baterai, dan lain-lain. Selain itu, bisa juga dikembangkan untuk pembuatan aplikasi berbasis android yang terhubung dengan domain web *Charging Station* secara langsung.

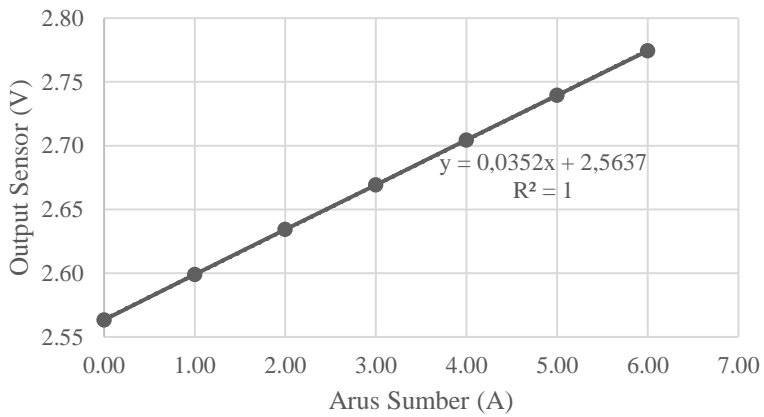
[Halaman ini sengaja dikosongkan]

LAMPIRAN

Tabel A.1 uji coba linearisasi sensor arus LA 55-P

Arus Sumber (A)	Output Sensor (V)
0	2,56
1	2,60
2	2,63
3	2,67
4	2,70
5	2,74
6	2,77

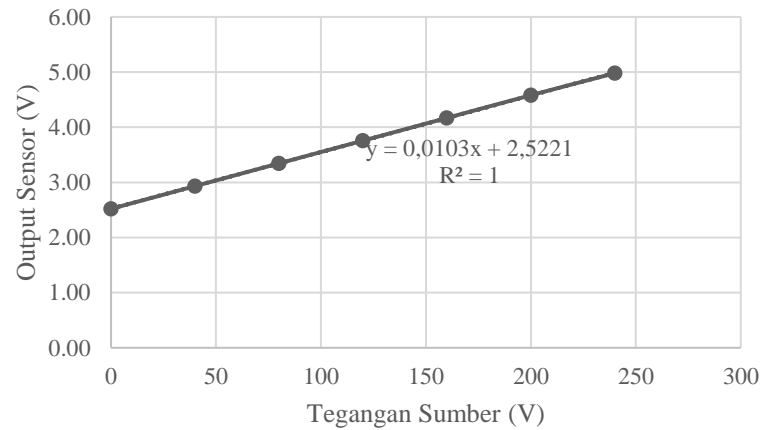
Gambar A.1 grafik uji coba linearisasi sensor arus LA 55-P



Tabel A.2 uji coba linearisasi sensor tegangan ZMPT 101B

Tegangan Sumber (V)	Output Sensor (V)
0	2,52
40	2,93
80	3,34
120	3,76
160	4,17
200	4,58
240	4,98

Gambar A.2 grafik uji coba linearisasi sensor tegangan ZMPT 101B



BAB VI

KESIMPULAN DAN SARAN

1.1 Kesimpulan

Dari hasil pengamatan dan percobaan selama perancangan, implementasi, dan uji coba aplikasi, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem Monitoring *Charging Station* menggunakan *Arduino Mega 2560* dengan *Ethernet Shield* sebagai modul jaringan internet yang tersambung dengan sensor-sensor yang ditampilkan pada *Heroku Cloud Platform*.
2. Pada pembacaan sensor arus dan tegangan AC dilakukan teknik sampling dengan mengambil arus dan tegangan tertinggi dan untuk arus dan tegangan DC dengan cara menghitung arus rata-rata pada sampling tersebut dengan error rata-rata di bawah 5 %.
3. Sistem Monitoring *Charging Station* Menggunakan *Solid State Relay* yang disambungkan pada *Arduino* berfungsi untuk melakukan kontrol dengan cara menekan tombol ON/OFF sebagai trigger *Arduino Processing* untuk memutuskan arus dan tegangan pada obyek yang akan diukur.

1.2 Saran

Berdasarkan perancangan dan pengujian sistem oleh penulis, maka untuk pengembangan selanjutnya dari sistem monitoring dan kontrol ini adalah melengkapi fitur-fitur yang ditampilkan pada web, seperti irradiant, suhu baterai, SOC baterai, dan lain-lain. Selain itu, bisa juga dikembangkan untuk pembuatan aplikasi berbasis android yang terhubung dengan domain web *Charging Station* secara langsung.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] N. Xiaoyong, Online Monitoring and Integrated Analysis System for EV Charging Station, IEEE, 2013.
- [2] L. Quishuo, T. Shun, X. Xiangning and W. Jianfeng, Monitoring and Analysis of Power Quality in Electric Vehicle Charging Station, IEEE, 2013.
- [3] G.Solution, "Cloud Services," 1 december 2015. [Online]. Available: <http://synegy.gs/Solutions/cloudServices>.
- [4] Prabowo, Wiratno, Hantoro, Asfani , "Infrastruktur Charging Station," Mobil Listrik Nasional, 2014.
- [5] F. Djuandi, "Pengenalan Arduino," ANDI, Mei 2015.
- [6] LEM, "Current Tranducers LA 55-P," LEM, November 2014.
- [7] Interplus Industry, "Single Phase AC Sensor," March 2016. [Online]. Available: <http://www.interplus-industry.com>.
- [8] Resistor Guide, "Shunt Resistor," 2 Maret 2016. [Online]. Available: <http://www.resistorguide.com/shunt-resistor/>.
- [9] Wikipedia, "volatage divider tutorial," 10 Maret 2106. [Online]. Available: http://en.wikipedia.org/wiki/Voltage_divider/.
- [10] Y. Rakhmatullah, "Solid State Relay," Universitas Airlangga Surabaya, 2016.
- [11] TP LINK, "Router TP LINK," Maret 2016. [Online]. Available:<http://www.tp-link.co.id/products/?categoryid=202>.
- [12] K. Tatroe, "Programming PHP Third Edition," USA, O'Reilly Media Inc, 2013.
- [13] O.W.Porbo, "Cloud Computing Server," Yogyakarta, ANDI, 2012.
- [14] Ashari Juang, "Pengenalan Heroku Cloud Application Platform," 13 Maret 2016. [Online]. Available: <http://www.asharijuang.com/pengenalan-heroku/>

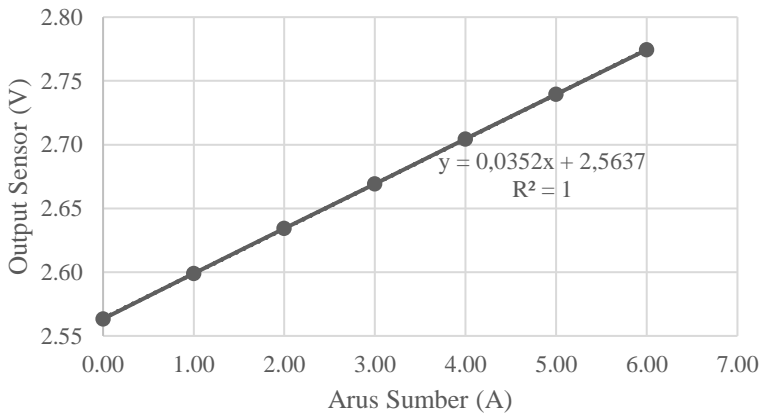
[Halaman ini sengaja dikosongkan]

LAMPIRAN

Tabel A.1 uji coba linearisasi sensor arus LA 55-P

Arus Sumber (A)	Output Sensor (V)
0	2,56
1	2,60
2	2,63
3	2,67
4	2,70
5	2,74
6	2,77

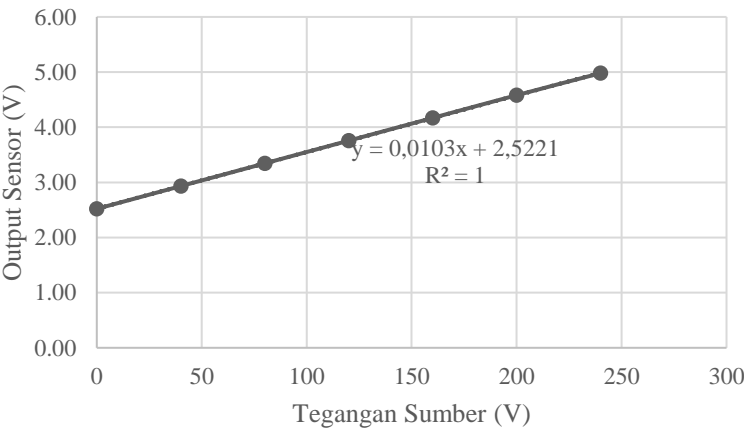
Gambar A.1 grafik uji coba linearisasi sensor arus LA 55-P



Tabel A.2 uji coba linearisasi sensor tegangan ZMPT 101B

Tegangan Sumber (V)	Output Sensor (V)
0	2,52
40	2,93
80	3,34
120	3,76
160	4,17
200	4,58
240	4,98

Gambar A.2 grafik uji coba linearisasi sensor tegangan ZMPT 101B



BIODATA PENULIS



Muhammad Farid Wajdi, lahir di Jakarta pada 27 Juni 1994. Penulis menempuh pendidikan mulai dari TK Cut Nyak Dhien Tangerang (1999 - 2000). SD Sunan Bonang Tangerang (2000 - 2006), SMPN 1 Kota Tangerang (2006 - 2009), SMA Negeri Cahaya Madani Banten Boarding School (2009 - 2012) dan S1 Teknik Informatika ITS (2012 - 2016).

Selama masa kuliah, penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer (HMTTC). Diantaranya adalah menjadi staff departemen pengembangan profesi HMTTC 2013 – 2016.. Penulis juga aktif dalam kegiatan kepanitiaan Schematics sebagai staff NST Schematics 2013. Penulis aktif mengikuti beberapa pelatihan sertifikasi diantaranya web program, web design, Huawei Certified Network Associate student 2016

Selama kuliah di teknik informatika ITS, penulis mengambil rumpun mata kuliah Komputasi Berbasis Jaringan (KBJ). Komunikasi dengan penulis dapat melalui email : **faridwajdi27@gmail.com**.